Online Evaluation of Audiences for Targeted Advertising via Bandit Experiments

Tong Geng,[1] **Xiliang Lin,**[1] **Harikesh S. Nair**[2]
[1]JD.com
[2]JD.com and Stanford University
{tong.geng, xiliang.lin}@jd.com, harikesh.nair@stanford.com

Abstract

Firms implementing digital advertising campaigns face a complex problem in determining the right match between their advertising creatives and target audiences. Typical solutions to the problem have leveraged non-experimental methods, or used "split-testing" strategies that have not explicitly addressed the complexities induced by targeted audiences that can potentially overlap with one another. This paper presents an adaptive algorithm that addresses the problem via online experimentation. The algorithm is set up as a contextual bandit and addresses the overlap issue by partitioning the target audiences into disjoint, non-overlapping sub-populations. It learns an optimal creative display policy in the disjoint space, while assessing in parallel which creative has the best match in the space of possibly overlapping target audiences. Experiments show that the proposed method is more efficient compared to naive "split-testing" or non-adaptive "A/B/n" testing based methods. We also describe a testing product we built that uses the algorithm. The product is currently deployed on the advertising platform of JD.com, an eCommerce company and a publisher of digital ads in China.

1 Introduction

A critical determinant of the success of advertising campaigns is picking the right audience to target. As digital ad-markets have matured and the ability to target advertising has improved, the range of targeting options has expanded, and the profile of possible audiences have become complex. Both advertisers and publishers now rely on data-driven methods to evaluate audiences and to find effective options with which to advertise to them. This paper presents a new bandit algorithm along with a product built to facilitate such evaluations via online experimentation.

The problem addressed is as follows. An advertiser designing a campaign wants to pick, from a set of possible target audiences and creatives, a *creative-target audience combination* that provides her the highest expected payoff in the campaign. The target audiences can be complex, potentially overlapping with each other, and the creatives can be any type of media (picture, video, text etc). We would like to design an experiment to find the best creative-target audience combination for the advertiser while minimizing her costs of experimentation.

When *only* creatives have to be compared to each other, the typical practice is to leverage an "A/B/n" experimental design in which creatives represent arms, so that the best creative is found by ranking the expected payoffs for users randomized into the arms. When target audiences have to be evaluated in addition, extending this design – treating creative-target audience combinations as arms – is problematic. The main difficulty is possible overlap in the target audiences that are compared (e.g., "San Francisco users" and "Male users"). This complicates user assignment because it is not obvious to which constituent arm, a user belonging to an overlapping region should be assigned (e.g., should a Male user from San Francisco be assigned to the "San Francisco-creative" arm or the "Male-creative" arm?). Assigning the overlapping user to one of the constituent arms violates the representativeness of the arms (e.g., if we use a rule that Male users from San Francisco will always be assigned to the "San Francisco-creative" arm, the "Male-creative" arm will have no San Franciscans, and will not represent the distribution of Male users in the platform population).[1] Such assignment also under-utilizes data: though the feedback from the user is informative of all constituent arms, it is being used to learn the best creative for only one picked arm (e.g., if we assign a Male user from San Francisco to the "San Francisco-creative" arm, we do not learn from him the value of the "Male-creative" arm, though his behavior is informative of that arm).

Another difficulty is that typical "A/B/n" test designs keep the sample/traffic splits constant as the test progresses. Therefore, both good and bad creatives will be allocated the same amount of traffic during the test. Instead, as we learn during the test that an arm is not performing well, reducing its traffic allocation can reduce the cost of experimentation.

The goal of this paper is to develop an algorithm that addresses these issues. It has two broad steps. In step one, we split the compared target audiences (henceforth "*TA*"s) into disjoint audience sub-populations (henceforth "*DA*"s), so the set of *DA*s fully span the set of *TA*s. In step two, we train a bandit with the creatives as arms, the payoffs to the

[1] Random assignment of users in overlapping regions to parent arms does not solve the issue; discussed later in the paper.

advertiser as rewards, and the *DA*s, rather than the *TA*s as the contexts. As the test progresses, we aggregate over all *DA*s that correspond to each *TA* to adaptively learn the best creative-*TA* match. In essence, we learn an optimal creative allocation policy at the disjoint sub-population level, while making progress towards the test goal at the *TA* level. Because the *DA*s have no overlap, each user can be mapped to a distinct *DA*, addressing the assignment problem. Because all *DA*s that map to a *TA* help inform the value of that *TA*, learning is also efficient. Further, tailoring the bandit's policy to a more finely specified context $-$ i.e., the *DA* $-$ allows it to match the creative to the user's tastes more finely, thereby improving payoffs and reducing expected regret, while delivering on the goal of assessing the best combination at the level of a more aggregated audience. The adaptive nature of the test ensures the traffic is allocated in a way that reduces the cost to the advertiser from running the test, because creatives that are learned to have low value early are allocated lesser traffic within each *DA* as the test progresses. The overall algorithm is implemented as a contextual Thompson Sampler (henceforth "TS"; see (Russo et al. 2018) for an overview).

Increasing the overlap in the tested *TA*s increases the payoff similarity between the *TA*s, making it harder to detect separation. An attractive feature of the proposed algorithm is that feedback on the performance of *DA*s helps inform the performance of all *TA*s to which they belong. This *cross-audience learning* serves as a counterbalancing force that keeps performance stable as overlap increases, preventing the sample sizes required to stop the test from growing unacceptably large and making the algorithm impractical.

In several simulations, we show the proposed TS performs well in realistic situations, including with high levels of overlap; and is competitive against benchmark methods including non-adaptive designs and "split-testing" designs currently used in industry. To illustrate real-world performance, we also discuss a case-study from a testing product on the advertising platform of `JD.com`, where the algorithm is deployed.

2 Related Work and Other Approaches

There is a mature literature on successful applications of bandits in web content optimization (e.g., (Agarwal, Chen, and Elango 2009), (Li et al. 2010), (Chapelle and Li 2011), (Urban et al. 2014), (Agarwal et al. 2016)). This paper belongs to a sub-stream of this work on using bandits for controlled experiments on the web. The closest papers to our work are the complementary papers by (Scott 2015), (Schwartz, Bradlow, and Fader 2017) and (Ju et al. 2019) who propose using bandit experiments to evaluate creatives for targeted advertising, without focusing on the problem addressed here of comparing target audiences.

In industry, a popular experimental design to compare *TA*s for advertising campaigns is "audience split-testing" (e.g., (Facebook 2019), (Tencent 2019)). Suppose there is only one creative, and K *TA*s are to be compared. The audience split-testing design randomizes test users into K arms, each of which is associated with the same creative, but which correspond respectively to the K *TA*s. Conditional on being randomized into an arm, a user is shown the creative only if his features match the arms' *TA* definition. This ensures that the mix of overlapping and non-overlapping audiences is representative; however, the design under-utilizes the informational content of experimental traffic as there is no learning from users who are randomized into a test-arm but do not match its *TA* definition. Also, in contrast to the design proposed here, there is no cross-audience learning from overlapping users. In addition, the typical implementation of split-testing is non-adaptive, and is not cost minimizing unlike the adaptive design presented here.

A possible strategy for maintaining the representativeness of *TA*s in the test is to randomly allocate some proportion p of users in each overlapping region to the *TA*s the region overlaps with. Unfortunately, no value of p exists that maintains representativeness after such allocation while retaining all the data. To illustrate, suppose we have two *TA*s ($TA1$ and $TA2$) that overlap with each other, so we have three *DA*s, $DA1$, $DA2$ and $DA3$, with $DA2$ belonging to both $TA1$ and $TA2$. Suppose in the test, a representative sample of N_{DA1}, N_{DA2}, and N_{DA3} users belonging to each of the three *DA*s arrive, and have to be assigned in this manner to $TA1$ and $TA2$. If we allocate proportion p of users in $DA2$ to $TA1$, the proportion of $DA2$ users in $TA1$ is $P(DA2|TA1) = \frac{p \times N_{DA2}}{p \times N_{DA2} + N_{DA1}}$. However, to be representative of the population, we need this proportion to be $\frac{N_{DA2}}{N_{DA2} + N_{DA1}}$. The only value of p that makes $TA1$ under this scheme representative is 1. However, when $p = 1$, the proportion of $DA2$ in $TA2$ is 0, making $TA2$ under this scheme not representative of $TA2$ in the population. One can restore representativeness by dropping a randomly picked proportion $1 - p$ of N_{DA1} users and p of N_{DA2} users. But this involves throwing away data and induces the same issue as the "audience split-testing" design above of under-utilizing the informational content of experimental traffic.

3 Method

The test takes as input $\mathbb{K} = \{1, .., K\}$ possible *TA*s and $\mathbb{R} = \{1, .., R\}$ creatives the advertiser wants to evaluate for her campaign. In step 1, we partition the users in the K *TA*s into a set $\mathbb{J} = \{1, .., J\}$ of J *DA*s. For example, if the *TA*s are "San Francisco users" and "Male users," we create three *DA*s, "San Francisco users, Male," "San Francisco users, Not Male," and "Non San Francisco users, Male."

In step 2, we treat each *DA* as a context, and each creative as an arm that is pulled adaptively based on the context. When a user i arrives at the platform, we categorize the user to a context based on his features, i.e., $i \in DA(j)$ if i's features match the definition of j, where $DA(j)$ denotes the set of users in DA j. A creative $r \in \mathbb{R}$ is then displayed to the user based on the context.

The cost of displaying creative r to user i in context j is denoted as b_{irj}. After the creative is displayed, the user's action, y_{irj}, is observed. The empirical implementation of the product uses clicks as the user feedback for updating the bandit, so y is treated as binary, i.e., $y_{irj} \in \{0, 1\}$. The payoff to the advertiser from the ad-impression, π_{irj}, is defined as π_{irj} $=\gamma \cdot y_{irj} - b_{irj}$, where γ is a factor that converts the user's action to monetary units. The goal of the bandit is to find an optimal policy $g(j) : \mathbb{J} \rightarrow \mathbb{R}$ which allocates the creative with the maximum expected payoff to a user with context j.

Thompson Sampling To develop the TS, we model the outcome y_{irj} in a Bayesian framework, and let

$$y_{irj} \sim p(y_{irj}|\theta_{rj}); \text{and, } \theta_{rj} \sim p(\theta_{rj}|\Omega_{rj}). \tag{1}$$

where θ_{rj} are the parameters that describe the distribution of action y_{irj}, and Ω_{rj} are the hyper-parameters governing the distribution of θ_{rj}. Since y is Bernoulli distributed, we make the typical assumption that the prior on θ is Beta which is conjugate to the Bernoulli distribution. With $\Omega_{rj} \equiv (\alpha_{rj}, \beta_{rj})$, we model,

$$y_{irj} \sim \texttt{Ber}(\theta_{rj}); \text{and, } \theta_{rj} \sim \texttt{Beta}(\alpha_{rj}, \beta_{rj}). \tag{2}$$

Given $y_{irj} \sim \texttt{Ber}(\theta_{rj})$, the expected payoff of each creative-disjoint sub-population combination (henceforth "*C-DA*") is $\mu^{\pi}_{rj}(\theta_{rj}) = \mathbb{E}[\pi_{irj}] = \gamma\mathbb{E}[y_{irj}] - \mathbb{E}[b_{irj}] = \gamma\theta_{rj} - \bar{b}_{rj}, \forall r \in \mathbb{R}, j \in \mathbb{J}$, where $\bar{b}_{rj}$ is the average cost of showing the creative r to the users in $DA(j)$.[2]

To make clear how the bandit updates parameters, we add the index t for batch. Before the test starts, $t = 1$, we set diffuse priors and let $\alpha_{rj,t=1} = 1, \beta_{rj,t=1} = 1, \forall r \in \mathbb{R}, j \in \mathbb{J}$. This prior implies the probability of taking action y, $\theta_{rj,t=1}, \forall r \in \mathbb{R}, j \in \mathbb{J}$ is uniformly distributed between 0% and 100%.

In batch t, N_t users arrive. The TS displays creatives to these users dynamically, by allocating each creative according to the posterior probability each creative offers the highest expected payoffs given the user's context. Given the posterior at the beginning of batch t, the probability a creative r provides the highest expected payoff is,

$$w_{rjt} = Pr[\mu^{\pi}_{rj}(\theta_{rjt}) = \max_{r\in\mathbb{R}}(\mu^{\pi}_{rj}(\theta_{rjt}))|\vec{\alpha}_{jt}, \vec{\beta}_{jt}], \tag{3}$$

where $\vec{\alpha}_{jt} = [\alpha_{1jt}, \ldots, \alpha_{Rjt}]'$ and $\vec{\beta}_{jt} = [\beta_{1jt}, \ldots, \beta_{Rjt}]'$ are the parameters of the posterior distribution of $\vec{\theta}_{jt} = [\theta_{1jt}, \ldots, \theta_{Rjt}]'$.

To implement this allocation, for each user $i = 1, .., N_t$ who arrives in batch t, we determine his context j, and make a draw of the $R \times 1$ vector of parameters, $\tilde{\boldsymbol{\theta}}^{(i)}_{jt}$. Element $\tilde{\theta}^{(i)}_{rjt}$ of the vector is drawn from $\texttt{Beta}(\alpha_{rjt}, \beta_{rjt})$ for $r \in \mathbb{R}$. Then, we compute the payoff for each creative r as $\mu^{\pi}_{rj}(\tilde{\theta}^{(i)}_{rjt}) = \gamma\tilde{\theta}^{(i)}_{rjt} - \bar{b}_{rj}$, and display to i the creative with the highest $\mu^{\pi}_{rj}(\tilde{\theta}^{(i)}_{rjt})$.

We update all parameters at the end of processing the batch, after the outcomes for all users in the batch is observed. We compute the sum of binary outcomes for each *C-DA* combination as $s_{rjt} = \sum_{i=1}^{n_{rjt}} y_{irjt}, \forall r \in \mathbb{R}, j \in \mathbb{J}$, where n_{rjt} is the number of users with context j allocated to creative r in batch t. Then, we update parameters as $\vec{\alpha}_{j(t+1)} = \vec{\alpha}_{jt} + \vec{s}_{jt}$ and $\vec{\beta}_{j(t+1)} = \vec{\beta}_{jt} + \vec{n}_{jt} - \vec{s}_{jt}, \forall j \in \mathbb{J}$, where $\vec{s}_{jt} = [s_{1jt}, \ldots, s_{Rjt}]'$, and $\vec{n}_{jt} = [n_{1jt}, \ldots, n_{Rjt}]'$.

Then, we enter batch $t + 1$, and use $\vec{\alpha}_{j(t+1)}$ and $\vec{\beta}_{j(t+1)}$ as the posterior parameters to allocate creatives at $t + 1$. We repeat this process until a pre-specified stopping condition (outlined below) is met.

Probabilistic Aggregation and Stopping Rule While the contextual bandit is set up to learn the best *C-DA* combination, the goal of the test is to learn the best creative-target audience combination (henceforth "*C-TA*"). As such, we compute the expected payoff of each *C-TA* combination by aggregating the payoffs of corresponding *C-DA* combinations, and stop on the basis of the regret associated with learning the best *C-TA* combination.

Using the law of total probability, we can aggregate across all *C-DA*s associated with *C-TA* combination (r, k) to obtain λ_{rkt},

$$\lambda_{rkt} = \sum_{j\in\mathcal{O}(k)} \theta_{rjt} \cdot \hat{p}(j|k). \tag{4}$$

In equation (4), λ_{rkt} is the probability that a user picked at random *from within TA(k)* in batch t, takes the action $y = 1$ upon being displayed creative r; $\hat{p}(j|k)$ is the probability (in the platform population) that a user belonging to $TA(k)$ is also of the context j; and $\mathcal{O}(k)$ is the set of disjoint sub-populations (js) whose associated *DA(j)*s are subsets of $TA(k)$.

Given equation (4), the posterior distribution of θ_{rjt}s from the TS induces a distribution of λ_{rkt}s. We can obtain draws from this distribution using Monte Carlo sampling. For each draw $\theta^{(h)}_{rkt}, h = 1, .., H$ from $\texttt{Beta}(\alpha_{rjt}, \beta_{rjt})$, we can use equation (4) to construct a corresponding $\lambda^{(h)}_{rkt}, h = 1, .., H$. For each such $\lambda^{(h)}_{rkt}$, we can similarly compute the implied expected payoff to the advertiser from displaying creative r to a user picked at random from within TA(k) in batch t,

$$\omega^{\pi}_{rkt}(\lambda^{(h)}_{rk}) \;\; = \;\; \gamma\lambda^{(h)}_{rkt} - \bar{b}_{rk}, \forall r \;\; \in \;\; \mathbb{R}, k \;\; \in \;\; \mathbb{K}, \tag{5}$$

where $\bar{b}_{rk}$ is the average cost for showing creative r to target audience k, which can be obtained by aggregating $\bar{b}_{rj}$ through analogously applying equation (4). Taking the H values of $\omega^{\pi}_{rkt}(\lambda^{(h)}_{rk})$ for each (r, k), we let r^*_{kt} denote the creative that has the highest expected payoff within each *TA* k across all H draws, i.e.,

$$r^*_{kt} = arg \max_{r\in\mathbb{R}} \max_{h=1,..,H} \omega^{\pi}_{rkt}(\lambda^{(h)}_{rk}). \tag{6}$$

Hence, $\omega^{\pi}_{r^*_{kt},kt}(\lambda^{(h)}_{rkt})$ denote the expected payoff for creative r^*_{kt} evaluated at draw h. Also, define $\omega^{\pi}_{*kt}(\lambda^{(h)}_{rkt})$ as the expected payoff for the creative assessed as the best for *TA* k in draw h itself, i.e.,

$$\omega^{\pi}_{*kt}(\lambda^{(h)}_{rkt}) = \max_{r\in\mathbb{R}} \quad \omega^{\pi}_{rkt}(\lambda^{(h)}_{rk}), \tag{7}$$

[2] γ may be determined from prior estimation or advertisers' judgment of the value attached to users' actions. γ is pre-computed and held fixed during the test. $\bar{b}_{rj}$ and $\hat{p}(j|k)$ (defined later) can be pre-computed outside of the test from historical data and held fixed during the test, or inferred during the test using a simple bin estimator that computes these as averages over the observed cost and user contexts data.

Following (Scott 2015), the value $\omega^{\pi}_{*kt}(\lambda^{(h)}_{rkt}) - \omega^{\pi}_{r^*_{kt},kt}(\lambda^{(h)}_{rkt})$ represents an estimate of the regret in batch t for *TA* k at draw h. Normalizing it by the expected payoff of the best creative across draws gives a unit-free metric of regret for each draw h for each *TA* k,

$$\rho^{(h)}_{kt} = \frac{\omega^{\pi}_{*kt}(\lambda^{(h)}_{rkt}) - \omega^{\pi}_{r^*_{kt},kt}(\lambda^{(h)}_{rkt})}{\omega^{\pi}_{r^*_{kt},kt}(\lambda^{(h)}_{rkt})}, \tag{8}$$

Let $pPVR(k,t)$ be the 95th percentile of $\rho^{(h)}_{kt}$ across the H draws. We stop the test when,

$$\max_{k\in\mathbb{K}} pPVR(k,t) < 0.01. \tag{9}$$

In other words, we stop the test when the normalized regret for all *TA*s we are interested in falls below 0.01.[3] Therefore, while we learn an optimal creative displaying policy for each *DA*, we stop the algorithm when we find the best creative for each *TA* in terms of minimal regret. Algorithm 1 shows the full procedure.

4 Experiments

This section reports on experiments that establish the face validity of the TS; compares it to audience split testing and a random allocation schema where each creative is allocated to each context with equal probability; and explores its performance when the degree of overlap in *TA*s increases.

For the experiments, we consider a setup with 2 creatives and 2 overlapping *TA*s, implying 3 *DA*s, 4 *C-TA* combinations and 6 *C-DA* combinations as shown in Figure (1). The *TA*s are assumed to be of equal sizes, with an overlap of 50%.[4] We set the display cost b_{irj} to zero and $\gamma = 1$ so we can work with the *CTR* directly as the payoffs (therefore, we interpret the cost of experimentation as the opportunity cost to the advertiser of not showing the best combination.) We simulate 1,000 values for the expected *CTR*s of the 6 *C-DA* combinations from uniform distributions (with supports shown in Figure (1)). Under these values, C_1-DA_1 has the highest expected *CTR* amongst the *C-DA* combinations, and C_1-TA_1 the highest amongst the *C-TA* combinations. We run the TS for each simulated value to obtain 1,000 bandit replications. For each replication, we update probabilities over batches of 100 observations, and stop the sampling

Algorithm 1 *TS* to identify best *C-TA* combination

```
$K$ TAs are re-partitioned into $J$ DAs
$t \leftarrow 1$
$\alpha_{rjt} \leftarrow 1, \beta_{rjt} \leftarrow 1, \forall r \in \mathbb{R}, j \in \mathbb{J}$
Obtain from historical data $\hat{p}(j|k), \gamma, \bar{b}_{rj}, \forall r \in \mathbb{R}, j \in \mathbb{J}, k \in \mathbb{K}$
$pPVR(k,t) \leftarrow 1, \forall k \in \mathbb{K}$
while $\max_{k\in\mathbb{K}} pPVR(k,t) < 0.01$ do
    A batch of $N_t$ users arrive
    for all users do
        Sample $\tilde{\theta}^{(i)}_{rjt}$ using $\mathtt{Beta}(\alpha_{rjt}, \beta_{rjt})$ for $r \in \mathbb{R}$
        Feed creative $I_{it} = argmax_{r\in\mathbb{R}}\gamma\tilde{\theta}^{(i)}_{rjt} - \bar{b}_{rjt}$
    end for
    Collect data $\{y_{irjt}\}_{i=1}^{N_t}, \{n_{rjt}\}_{r\in\mathbb{R},j\in\mathbb{J}}$
    Compute $s_{rjt} = \sum_{i=1}^{n_{rjt}} y_{irjt}, \forall r \in \mathbb{R}, j \in \mathbb{J}$
    Update $\alpha_{rj(t+1)} = \alpha_{rjt} + s_{rjt}, \forall r \in \mathbb{R}, j \in \mathbb{J}$
    Update $\beta_{rj(t+1)} = \beta_{rjt} + n_{rjt} - s_{rjt}, \forall r \in \mathbb{R}, j \in \mathbb{J}$
    Make $h = 1, .., H$ draws of $\theta_{rj(t+1)}$s, i.e.
        $\begin{bmatrix} \theta_{11(t+1)} \\ ... \\ \theta_{rj(t+1)} \\ ... \\ \theta_{RJ(t+1)} \end{bmatrix}^{(h)} \sim \begin{bmatrix} \mathtt{Beta}(\alpha_{11(t+1)}, \beta_{11(t+1)}) \\ ... \\ \mathtt{Beta}(\alpha_{rj(t+1)}, \beta_{rj(t+1)}) \\ ... \\ \mathtt{Beta}(\alpha_{RJ(t+1)}, \beta_{RJ(t+1)}) \end{bmatrix}^{(h)} , \forall h = 1, ..., H$
    Compute $\vec{\lambda}^{(h)}_{t+1} =$
        $\begin{bmatrix} \lambda_{11(t+1)} \\ ... \\ \lambda_{rk(t+1)} \\ ... \\ \lambda_{RK(t+1)} \end{bmatrix}^{(h)} = \begin{bmatrix} \sum_{j\in O(k=1)} \hat{p}(j|k=1) \cdot \theta_{rj(t+1)} \\ ... \\ \sum_{j\in O(k)} \hat{p}(j|k) \cdot \theta_{rj(t+1)} \\ ... \\ \sum_{j\in O(k=K)} \hat{p}(j|k=K) \cdot \theta_{rj(t+1)} \end{bmatrix}^{(h)} , \forall h = 1, ..., H$
    Compute $\vec{\omega^{\pi}}^{(h)}_{t+1}(\vec{\lambda}^{(h)}_{t+1}) =$
        $\begin{bmatrix} \omega^{\pi}_{11(t+1)} \\ ... \\ \omega^{\pi}_{rk(t+1)} \\ ... \\ \omega^{\pi}_{RK(t+1)} \end{bmatrix}^{(h)} = \begin{bmatrix} \gamma \cdot \lambda_{11(t+1)} - \bar{b}_{11(t+1)} \\ ... \\ \gamma \cdot \lambda_{rkt} - \bar{b}_{rk(t+1)} \\ ... \\ \gamma \cdot \lambda_{RKt} - \bar{b}_{RK(t+1)} \end{bmatrix}^{(h)} , \forall h = 1, ..., H$
    Set $\rho^{(h)}_{k(t+1)} = [\omega^{\pi}_{*k(t+1)}(\lambda^{(h)}_{rk(t+1)}) - \omega^{\pi}_{r^*_{k(t+1)},k(t+1)}(\lambda^{(h)}_{rk(t+1)})]/\omega^{\pi}_{r^*_{k(t+1)},k(t+1)}(\lambda^{(h)}_{rk(t+1)}), \forall h = 1, ..., H, k \in \mathbb{K}$
    $\forall k \in \mathbb{K}$, calculate $pPVR(k, t+1)$ as the 95th percentile across the $H$ draws of $\rho^{(h)}_{k(t+1)}$
    Set $t \leftarrow t + 1$
end while
```

[3] Other stopping rules may also be used, for example, based on posterior probabilities, or based on practical criteria that the test runs till the budget is exhausted (which protects the advertiser's interests since the budget is allocated to the best creative). The formal question of how to stop a TS when doing Bayesian inference is still an open issue. While data-based stopping rules are known to affect frequentist inference, Bayesian inference has traditionally been viewed as unaffected by optional stopping (e.g., (Edwards, Lindman, and Savage 1963)), though the debate is still unresolved in the statistics and machine learning community (e.g., (Rouder 2014) vs. (de Heide and Grünwald 2018)). This paper adopts a stopping rule reflecting practical product-related considerations, and does not address this debate.

[4] Specifically, $\Pr(TA_1) = \Pr(TA_2) = .5$; $\Pr(DA_1|TA_1) = \Pr(DA_2|TA_1) = 0.5$; $\Pr(DA_2|TA_2) = \Pr(DA_3|TA_2) = 0.5$; and $\Pr(DA_1|TA_2) = \Pr(DA_3|TA_1) = 0$.

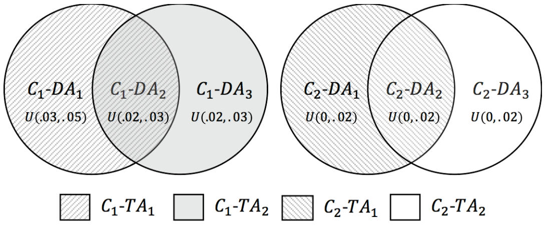
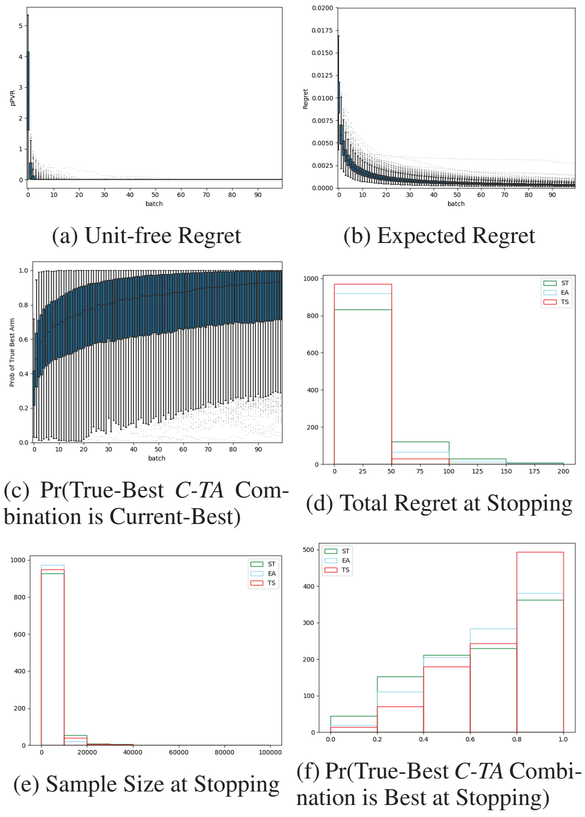


Figure 1: Simulation Setup: 2 Cs, 2 TAs and 3 DAs



(a) Unit-free Regret

(b) Expected Regret

(c) Pr(True-Best *C-TA* Combination is Current-Best)

(d) Total Regret at Stopping

(e) Sample Size at Stopping

(f) Pr(True-Best *C-TA* Combination is Best at Stopping)

Figure 2: Results from 1,000 Replications for TS and Comparisons to Equal Allocation and Split-Testing

when we have 1000 batches of data. Then, we report in Figure (2), box-plots across replications of the performance of the TS as batches of data are collected, plotting these at every 10th batch.

Figures (2a and 2b) plot the evolution over batches in the unit-free regret (*pPVR*) and the expected regret per impression, where the latter is defined as the expected clicks lost per impression in a batch when displaying a creative other than the true-best for each *DA*, evaluated at the true parameters.[5] If the TS progressively allocates more traffic to creatives with higher probability of being the best arm in each context (*DA*), the regret should fall as more data is accumulated. Consistent with this, both metrics are seen to fall as the number of batches increases in our simulation. The cutoff of 0.01 *pPVR* is met in 1,000 batches in all replications. Figure (2c) shows the posterior probability implied by TS in each batch that the true-best *C-TA* is currently the best.[6] The posterior is seen to converge to the true-best combination as more batches are sampled.

We now compare the proposed TS algorithm to an Equal Allocation algorithm (henceforth "EA") and a Split-Testing algorithm (henceforth "ST"). EA is analogous to "A/B/n" testing in that it is non-adaptive: the allocation of traffic to creatives for each *DA* is held fixed, and not changed across batches. Instead, in each batch, we allocate traffic equally to each of the $r \in \mathbb{R}$ creatives for each *DA*. ST follows the design described in §2, and traffic is allocated at the level of *C-TA* (rather than *C-DA*) combinations. Each user is assigned randomly with fixed, equal probability to one of $R \times K$ *C-TA* arms (4 in this simulation), and a creative is displayed only if a user's features match the arm's *TA* definition.

To do the comparison, we repeat the same 1,000 replications as above with the same configurations, but this time stop each replication when the criterion in equation (9) is reached. In other words, for each of TS, EA and ST algorithms, we maintain a posterior belief about the best *C-TA* combination, which we update after every batch.[7] In TS, the traffic allocation reflects this posterior adaptively, while in EA and ST, the traffic splits are held fixed; and the same stopping criteria is imposed in both. All parameters are held the same.

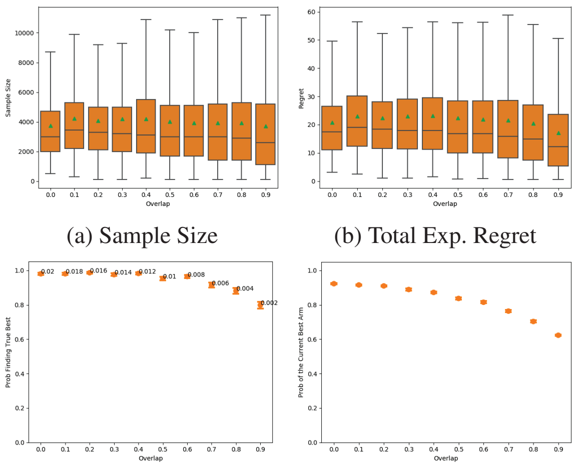
Figure (2d) shows that TS generates the smallest amount of expected regret, and the sample sizes required to exit the experiments under TS are between those under EA and those under ST (Figure (2e)). This is because the expected regret per impression under EA and ST remains constant over batches, while as Figure (2b) demonstrated, the expected regret per impression under TS steadily decreases as more batches arrive. ST generates the most regret and requires the largest sample sizes, since it is not only non-adaptive, but also discards a portion of the traffic and the information that could have been gained from this portion. Figure 2f shows that the TS puts more mass at stopping on the true-best *C-TA* combination compared to EA and ST. Across replications, this allows TS to correctly identify the true-best combination 85.8% of the time at stopping, compared to 77.8% for EA and 70.8% for ST. Overall, the superior performance of the TS relative to EA are consistent with the experiments reported in (Scott 2010).

Next, we assess how the extent to which audiences overlap affects performance. This demonstrates the cross-audience learning effect in the algorithm. To do this, we fix the *CTR*s of the six *C-DA* combinations C_1-DA_1, C_2-DA_1, C_1-DA_2, C_2-DA_2, C_1-DA_3, C_2-DA_3 to be [.01,.03,.03,.05,.025,.035]. We vary the size of the overlapped audience, i.e. $\Pr(DA_2|TA_1) = \Pr(DA_2|TA_2)$, on

[5] Specifically, the expected regret per impression in each batch t is $\sum_{k\in\mathbb{K}}\sum_{j\in O(k)}\hat{p}(j|k)\sum_{r\in\mathbb{R}} w_{rjt}(\theta_{rj}^{\text{true}} - \max_{r\in\mathbb{R}}\theta_{rj}^{\text{true}})$.

[6] Note, these probabilities are not the same as the distribution of traffic allocated by the TS, since traffic is allocated based on *DA* and not *TA*.

[7] Note that, we do not need to partition the *TA*s under ST, and instead directly set up the model at the *C-TA* level under ST.



(a) Sample Size (b) Total Exp. Regret

(c) Prop. of True-Best Found (d) Post. Prob of Best at Stop

Figure 3: TS Performance with Increasing Overlap

a grid from 0-$.9$. For each grid value, we run the TS for 1,000 replications, taking the 6 *C-DA CTR*s as the truth, stopping each replication per equation (9). We then present in Figure 3 box-plots across these replications as a function of the degree of overlap. Along the x-axis, the two target audiences become more similar, increasing cross-audience learning, but decreasing their payoff differences.

Figures (3a and 3b) show that sample sizes required for stopping and total expected regret per impression remain roughly the same as overlap increases, suggesting the two effects largely cancel each other. Figure (3c) shows the proportion of 1,000 replications that correctly identify the true-best *C-TA* combination as the best at stopping. The annotations label the payoff difference in the top-2 combinations, showing the payoffs become tighter as the overlapping increases. We see that the TS works well for reasonably high values of overlap, but as the payoff differences become very small, it becomes difficult to correctly identify the true-best *C-TA* combination. Figure (3d) explains this pattern by showing the posterior probability of the best combination identified at stopping also decreases as the payoff differences grow very small. Finally, the `arXiv` version of the paper presents additional experiments that show that the observed degradation in performance of the TS at very high values of overlap disappears in a pure cross-audience learning setting.

Overall, these simulations suggest the proposed TS is viable in identifying best *C-TA* combinations for reasonably high levels of *TA* overlap. If the sampler is to be used in situations with extreme overlap, it may be necessary to impose additional conditions on the stopping rule based on posterior probabilities, in addition to the ones based on $pPVR$ across contexts in equation (9). This is left for future research.

5 Deployment

We designed an experimentation product based on algorithm. To use the product, an advertiser starts by setting up a test ad-campaign on the product system. The test campaign is similar to a typical ad-campaign, involving rules for bidding, budget, duration etc. The difference is that the advertiser defines K *TA*s and binds R creatives to the test-campaign, rather than one as typical; and the allocation of creatives to a user impression is managed by the TS algorithm. Both K and R are limited to a max of 5. Because the algorithm disjoints *TA*s, the number of contexts grows combinatorially as K increases, and this restriction keeps the total combinations manageable.

When a user arrives at `JD.com`, the ad-serving system retrieves the user's features. If the features activate the tag(s) of any of the K *TA*s, and satisfies the campaign's other requirements, the TS chooses a test creative according to the adaptively determined probability, and places a bid for it into the platform's auction system. The bids are chosen by the advertiser, but are required to be the same for all creatives in order to keep the comparison fair. The auction includes other advertisers who compete to display their creatives to this user. The system collects data on the outcome of the winning auctions and whether the user clicks on the creative when served; updates parameters every 10 minutes; and repeats this until the stopping criterion is met and the test is stopped. The data are aggregated and relevant statistical results regarding all the *C-TA* combinations are delivered to the advertiser. See `https://jzt.jd.com/gw/dissert/jzt-split/1897.html` for a product overview.

We discuss a case-study based on a test on the product. Though several other tests exhibit similar patterns, there is no claim this case-study is representative: we picked it so it illustrates well for the reader some features of the test environment and the performance of the TS.The test involves a large cellphone manufacturer. The advertiser set up 2 *TA*s and 3 creatives. The 2 *TA*s overlap, resulting in 3 *DA*s. Figure (4) shows the probability that each *C-TA* combination is estimated to be the best as the test progresses. The 6 possible combinations are shown in different colors and markers. During the initial 12 batches, the algorithm identifies the "*" and "+" combinations to be inferior and focuses on exploring the other 4 combinations. Then, the yellow "." combination starts to dominate the others and is finally chosen as the best. The advantage of the adaptive design is that most of the traffic during the test is allocated to *C-DA* combinations corresponding to the yellow "." combination, so the advertiser does not unnecessarily waste resources on assessing those that were learned to be inferior early on.

The test lasted about 6 hours with a total of 18,499 users and 631 clicks. The estimated *CTR*s of the six *C-TA* combinations C_1-TA_1, C_2-TA_1, C_3-TA_1 (yellow "." combination), C_1-TA_2, C_2-TA_2, C_3-TA_2 at stopping are [.028,.034,.048,.028,.017,.036]. Despite the short time span, the posterior probability induced by the sampling on the yellow "." combination being the best is quite high (98.4%). We use a back-of-the-envelope calculation to assess the economic efficiency of TS relative to EA in this test. Using the data, we simulate a scenario where we equally allocate across the creatives the same amount of traffic as this test used. We find TS generates 52 more clicks (8.2% of total clicks) than EA.

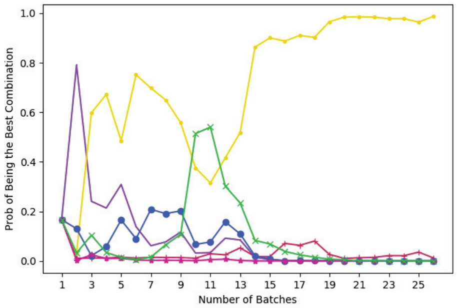


Figure 4: Results from Practical Implementation

In other tests, we found the product performs well even in situations where the creatives are quite similar and K, R are close to 5, without requiring unreasonable amounts of data or test time so as to make it unviable. Scaling the product to allow for larger sets of test combinations is a task for future research and development.

6 Conclusion

An adaptive algorithm to identify the best combination among a set of advertising creatives and *TA*s is presented. Experiments show that the proposed method is more efficient compared to naive "split-testing" or non-adaptive "A/B/n" testing based methods. The approach assumes that creatives do not induce long-term dependencies, for instance, that they do not affect future user arrival rates, and that auctions are unrelated to each other, for instance due to the existence of a binding budget constraint. These assumptions justify framing the problem as a multi-armed bandit, and could be relaxed by using a more general reinforcement learning framework.

References

Agarwal, A.; Bird, S.; Cozowicz, M.; Hoang, L.; Langford, J.; Lee, S.; Li, J.; Melamed, D.; Oshri, G.; Ribas, O.; Sen, S.; and Slivkins, A. 2016. Making contextual decisions with low technical debt. *arXiv:1606.03966*.

Agarwal, D.; Chen, B.; and Elango, P. 2009. Explore/exploit schemes for web content optimization. In *ICDM*, 1–10.

Chapelle, O., and Li, L. 2011. An empirical evaluation of thompson sampling. In *NIPS 2011*, 2249–2257.

de Heide, R., and Grünwald, P. D. 2018. Why optional stopping is a problem for Bayesians. *arXiv preprint arXiv:1708.08278*.

Edwards, W.; Lindman, H.; and Savage, L. J. 1963. Bayesian statistical inference for psychological research. *Psychological Review* 70(3):193–242.

Facebook. 2019. Split Testing & Test and Learn: About Split Testing. https://www.facebook.com/business/help/1738164643098669.

Ju, N.; Hu, D.; Henderson, A.; and Hong, L. 2019. A sequential test for selecting the better variant: Online A/B testing, adaptive allocation, and continuous monitoring. In *WSDM 2019*, 492–500.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *WWW '10*, 661–670.

Rouder, J. N. 2014. Optional stopping: No problem for Bayesians. *Psychonomic Bulletin & Review* 21(2):301–308.

Russo, D. J.; Van Roy, B.; Kazerouni, A.; Osband, I.; and Wen, Z. 2018. A tutorial on Thompson sampling. *Foundations and Trends® in Machine Learning* 11(1):1–96.

Schwartz, E. M.; Bradlow, E. T.; and Fader, P. S. 2017. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science* 36(4):500–522.

Scott, S. L. 2010. A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry* 26(6):639–658.

Scott, S. L. 2015. Multi-armed bandit experiments in the online service economy. *Applied Stochastic Models in Business and Industry* 31(1):37–45.

Tencent. 2019. Tencent advertising effects split comparison experiment tool online. https://mp.weixin.qq.com/s/jj-l6NTEzxEYyB5Ic4Wq7w.

Urban, G. L.; Liberali, G. G.; MacDonald, E.; Bordley, R.; and Hauser, J. R. 2014. Morphing banner advertising. *Marketing Science* 33(1):27–46.