Selecting Multiple Web Adverts - a Contextual Multi-armed Bandit with State Uncertainty

Abstract

We present a method to solve the problem of choosing a set of adverts to display to each of a sequence of web users. The objective is to maximise user clicks over time and to do so we must learn about the quality of each advert in an online manner by observing user clicks. We formulate the problem as a novel variant of a contextual combinatorial multi-armed bandit problem. Critical features of our formulation are that the context takes the form of a probability distribution over the user's latent topic preference, and that the rewards are a particular nonlinear function of the selected set and the context. Combined, these features ensure that optimal sets of adverts are appropriately diverse. We give a flexible solution method in which submodular optimisation is combined with existing bandit index policies. However, user state uncertainty creates ambiguity in interpreting user feedback which prohibits exact Bayesian updating, but we give an approximate method that is shown to work well. An algorithm base on Thompson sampling is tested using simulations and is shown to learn and perform effectively.

Keywords: multi-armed bandits, contextual bandits, statistical learning, diverse recommendation

1 Introduction

We present a contextual combinatorial multi-armed bandit method designed to solve the problem of choosing an appropriately diverse set of adverts to display to each of a sequence of users of a website. Our objective is to maximise the number of times that adverts are clicked on. To achieve this it is necessary to match a set of adverts to each user's interests. Information about user preferences is considered as a *context*, provided before the adverts for that user are selected. This user information will be based upon historical behaviour and recent activity (such as search terms). In contrast, information about the quality and relevance of adverts can only be learned by experimenting in a *bandit* framework, so that only by displaying an advert can any information be gained about it. Furthermore, due to uncertainties about both the user's preferences, and the adverts' relevances, it is necessary to select a *diverse set* of adverts to display to maximise the chance that at least one of the adverts matches the user's preferences sufficiently well to receive a click.

Consider for example a web user searching for “bicycle”. This user could be interested in many different types of bicycle (such as a mountain, racing or child’s) and will have different price preferences. The type- and price-preferences of the user are likely to be somewhat known based on previous web behaviour. On the other hand, with an ever-changing pool of bicycle-relevant adverts it is likely that the relevance of any particular advert to a particular type of bicycle will be uncertain and needs to be learned. Adverts of an inappropriate type are likely to be ignored while good quality adverts which match user preferences may perhaps be clicked.

The challenge of learning online the user response to the presented adverts is that posed by the *multi-armed bandit problem* (MAB). *Arms* (equivalent to adverts) are chosen in sequence with a *reward* (here a click or no click) received after each choice. By observing rewards we learn about the arms’ reward distributions which can then be used to inform which arms are chosen in future. The difficulty in the problem lies in how best to trade-off choosing arms for expected immediate reward given current information (exploitation) against choosing arms to learn about them and so to gain improved reward in the long term (exploration). Our multiple adverts problem extends the classical MAB since (i) we choose several arms at each time instead of one, (ii) at most one advert is clicked by the user at each time so the reward is a function of the *set* of arms not individual arms and, (iii) the reward depends on the users’ preferences as well as the selected arm set.

The dependence of reward on each user means we have a form of *contextual MAB* (e.g. Li et al. 2010, May et al. 2012) where the reward is a function of some *context* given at each time prior to choosing an arm. In the general contextual MAB the context is rarely given an interpretation or constraint. In contrast we assume the contexts follow the model of Edwards & Leslie (2018); each user is assumed to have a latent preference or *state*, corresponding to which topic of advert a user might click on, and the context is a probability mass function over this discrete set of topics, called a *topic preference vector*. The topic preference vector therefore represents the system’s beliefs about the user’s latent topic.

Rewards in the system are modelled as in Edwards & Leslie (2018). This model will be stated formally in Section 3 but the approach is based on using the topic preference vector in an informative manner. In particular, if the state was known then adverts could be chosen that are appropriate for that state. However with the state being latent the probability an advert is clicked must be averaged over the states. It may then be desirable to choose a set of adverts appropriate for several different states to ensure that there is a good chance of a click whichever of the possible states is true. Thus good advert sets tend to be *diverse* to avoid redundancy between similar adverts.

However in Edwards & Leslie (2018) all information about the adverts was assumed to be known and advert sets were chosen by solving a submodular optimisation problem; no learning was considered. A key innovation in this paper is that the quality and characteristics of the adverts can only be learned by observing user click behaviour. Furthermore, inference is challenging because we do not observe the true user preference or advert topics. An online expectation maximisation framework will be used to handle these latent user preferences. The inherently Bayesian nature of this framework enables prior information about the adverts to be utilised.

The main contributions of this paper are as follows. We give a formulation and solution method for our multiple advert problem with online learning of advert quality and topics. In doing this we:

- Formulate a contextual bandit problem which uses a context based on a latent user state such that the context is a probability distribution. This is a new form of context which allows for a richer representation of uncertainties involved in the contextual information. This is needed to accurately motivate the selection of diverse sets of objects but it introduces several difficulties that prohibit the easy use of standard Bayesian bandit methods.

- Develop a method for inference under user state uncertainty and bandit feedback which will have application beyond this problem. We adapt an online expectation maximisation algorithm for use in a Bayesian setting, test its performance and identify its limitations.

- Give a method by which existing bandit algorithms, which are designed for choosing single arms at a time, can be combined with submodular optimisation methods for choosing diverse sets of interacting elements. This allows uncertainty and learning to be incorporated into submodular optimisation.

- Present and test a complete policy using Thompson sampling paired with a sequential greedy set-choosing algorithm.

Related work will be discussed in Section 2. Section 3 will give a formal statement of the problem together with models for user click behaviour based on uncertainty about user preferences. Section 4 presents a Bayesian model for learning for when arm characteristics are not known which will be developed into a solution method in Section 5. This is tested in simulations in Section 6. Section 7 concludes with a summary of contributions and a discussion of issues.

The code used for all simulations in this paper can be found at `https://bitbucket.org/jedwards24/multiple_adverts`.

2 Related Work

The main features of our multiple adverts problem are: (i) A MAB where multiple arms (adverts) are chosen at each time; (ii) The arms interact so we need to learn about the reward of sets of arms rather than individual arms; (iii) The quality and characteristics (topics) of the arms must be learnt over time, (iv) There is a contextual aspect to the problem which comes from information about the users' preferences which are not known exactly; (v) Feedback from the user is limited to either a click on a single advert/arm or no click at all. This will be addressed within a Bayesian framework for inference and learning.

Several existing MAB frameworks allow selection of multiple arms at each time. In the *MAB with multiple plays* (e.g. Whittle 1988) rewards are a simple sum of the independent rewards of the individual arms which differs from our problem setup. The *combinatorial MAB* is also concerned with selecting sets of arms at each time. In Chen et al. (2013) rewards can be quite general functions of the arms selected but it does not include the reward formulation we give in Section 3 because of the dependence of rewards on the latent user state. They also assume that rewards from individual arms are observed while we address the more difficult problem where only the total reward is observed.

Another related framework is the *linear bandit* problem (e.g. Auer 2002), in which a weight vector is chosen at each time then we observe a reward which is some linear function of this vector. This has structural similarities with our work especially in the submodular setting of Yue & Guestrin (2011). However, the weight vectors (corresponding to adverts in our problem) are assumed known while the reward function (relating to the user) is constant over time and must be learned. In our problem we must learn about multiple adverts with a unique user at each time, which complicates the issue significantly as interactions create a combinatorial learning problem over the adverts *in addition* to user uncertainty.

Interactions within a set of objects has been studied in the area of *information or document retrieval*. The models for user click behaviour given in Section 3 build on ideas from Agrawal et al. (2009) and El-Arini et al. (2009). However, in both of these, as in related work in Yue & Guestrin (2011) and Radlinski et al. (2008), the quality and features of the available documents is assumed to be fixed and known. Streeter et al. (2009) shares a number of aspects of our problem with sets of arms having a similar reward structure. However, Yue & Guestrin (2011) notes that Streeter et al. (2009), as well as most similar set-based bandit work, assume a "feature-free model" (i.e. it does not utilise user contextual information).

A bandit problem with similar user uncertainty (but with only a single arm chosen at each time) can be found in Hauser et al. (2009) who studied adapting website designs based on users' cognitive styles. Their Bayesian updating approach is similar to that given here in Section 4.2. Their justification is heuristic while we come to the method via online expectation maximisation which gives stronger theoretical underpinnings. We also highlight circumstances where the method fails, which is relevant to the application in Hauser et al. (2009).

Schwartz et al. (2017) applies bandit learning to online advertising. Although there are features of their work relevant to ours, their model does not consider interactions between adverts displayed simultaneously which leads to a different modelling approach. Their problem is to select adverts with different attributes (e.g. size or message) to display on a range of websites. The hierarchical Bayesian model used incorporates advert attributes and models heterogeneity in the click through rate (CTR) across both adverts and the websites. Differences in advert CTRs across different websites represent differences between user populations of the websites and has similarities with our use of a user state. However, while the website on which the advert is placed is known and can be controlled, in our model the user state is uncontrolled, unobserved and cannot be learnt over time. This feature, which is crucial in the selection of advert sets, creates extra challenges in inference and the adaption of existing bandit algorithms.

Much work has been done in online search advertising on developing models for predicting user CTRs. Often these are built on standard models that are practical at the large scales of online applications, for example: logistic regression (McMahan et al. 2013, Richardson et al. 2007, Chapelle et al. 2015); Bayesian probit regression(Graepel et al. 2010); and linear Poisson regression (Chen et al. 2009). The feature spaces used for these usually take a non-specific form but several studies, such as Hillard et al. (2010) and Richardson et al. (2007), utilise the text of either the search term or keywords associated with the advert as features of their models. Similarly to our model Hillard et al. (2010) distinguishes between advert CTR and advert relevance - it is desirable to identify adverts that are relevant to the current user, which may be different to the one with the highest overall CTR. Chen et al. (2009) and Yan et al. (2009) use behavioural targeting to identify the adverts that are most relevant to users which incorporates user history into click prediction. The empirical study in Yan et al. (2009) found that the benefits of appropriate matching of adverts to users could be significant. In each of these cases there are similarities to our approach but none are designed to work with multiple interacting adverts which prevents their direct use in our work. In Section 7 we will discuss how our model for multiple adverts could be built upon to fit some of the frameworks used by the references given here.

The problem of web advertising includes a number of aspects that we do not consider directly here. We study the problem exclusively from the viewpoint of the publisher displaying the adverts and assume that the available adverts are given and that there are no constraints on their use. A related but different problem takes the view of an advertiser who must pay for their adverts to be displayed. Rusmevichientong & Williamson (2006) has most in common with our work. Here, the advertiser must bid for search advertising slots based on search keywords with a limited budget. Their method combines a stochastic knapsack with bandit learning.

In the area of operational research the majority of research into display advertising has been concerned with pricing and contracts between the publisher and the advertiser, mainly from the publisher's perspective. Traditionally, contracts may involve a price per view or per click with constraints on the number of times the advert is displayed. The publisher needs to meet the agreed constraints despite uncertainty in demand for slots, traffic, and click behaviour. Ahmed & Kwon (2014) considers the choice of contract size with pay-per-view pricing while the choice of price is studied using a queueing approach in Najafi-Asadolahi & Fridgeirsdottir (2014) and Fridgeirsdottir & Najafi-Asadolahi (2018) for pay-per-click and pay-per-view respectively. Hojjat et al. (2017) gives a framework which allows more complicated contracts where advertisers can specify how their adverts are displayed (e.g. advert placement and the demographics of targeted users). An alternative to fixed contracts that have become more popular recently are dynamic auctions where advertisers bid for slots offered by publishers. These are investigated by Balseiro et al. 2014) and Chen (2017) with the latter studying a market that includes both guaranteed contracts and dynamic auctions.

3 Problem Formulation

This section will formally state our multiple adverts problem. Here and in the rest of the paper we will use the terms arms and adverts interchangeably.

3.1 Topic Preference Vector

Our model for user click behaviour is based on the idea that each user has a latent preference or *state*. This state is never observed directly but we will assume that we are provided with a probability distribution over the user's states. This *topic preference vector* represents existing knowledge or beliefs about the current user's state.

The most general way to think about the state space is as a segmentation of the population where users with the same state will have similar click behaviour with regard to available adverts. We do not give details of how to choose appropriate segments or states here but many methods exist. For example, in the field of Recommender Systems users are often characterised by vectors of latent factors or features, corresponding to our topic preference vectors. Although these could be chosen using domain-specific knowledge, they can also be generated automatically by methods such as collaborative filtering (see, for example, Ricci et al. 2011). Our method does not put any constraints on the length of topic preference vectors so the number of states is also chosen as part of the feature generation process. Note that these states do not need to have any interpretation or meaning in order to be useful.

We characterise the arms by using weight vectors which correspond to the topic preference vector. The value in a given entry indicates how relevant the arm is to a user with that state (how appropriate the advert is to that topic). Adverts remain available over time so it is beneficial to learn the advert weights. Each user, however, is different from those that have been seen before so the topic preference vectors are unique to the current time so there is no learning about future users' states. Using a new, known topic preference vector at each time is not too strong an assumption since the time frame over which an advert is displayed will be small relative to the number of times a search term has been entered so much greater prior information would be available about searches (and therefore the user population's preferences) than about the adverts. We make no assumptions about being able to classify adverts in advance as being compatible with one state or another; this will be learnt over time. However if knowledge is available then this can be used in the priors for relevant weights.

The mathematical formulation of the users and arms is as follows. At each time step $t = 1, 2, \ldots$ a user arrives with a *state* $x_t \in \{1, \ldots, n\}$. This state is hidden but its distribution X_t is observed and is given by a topic preference vector $\mathbf{q}_t$ such that $\Pr(X_t = x) = q_{t,x}$. In response to this we present m arms as an (ordered) set $A_t \subseteq \mathcal{A}$ where $\mathcal{A}$ is the set of k available arms. The user will respond by selecting (or clicking) at most one arm. If any arm is clicked then a reward of one is received, with zero reward otherwise. A more general model would be to allow the reward given a click to vary between arms. For simplicity we do not do this here but the model and solution method can easily be adapted by substituting expected rewards for expected clicks.

Each arm $a \in \mathcal{A}$ is characterised by a weight vector $\mathbf{w}$ of length n with each $w_{a,x} \in (0, 1)$. Let $\mathbf{w}_A$ denote the set of vectors $\{\mathbf{w}_a\}, a \in A$. The weight $w_{a,x}$ represents the probability a user with state x will click advert a. Each weight is not known exactly but can be learnt over time as a result of the repeated selection of arms and observation of outcomes. The outcome at each time is given by the reward together with which advert (if any) is clicked. Further details on the feedback and learning process are given in Section 4.1.

This framework is complete if $m = 1$ advert is to be displayed and x is known: the click probability if a is presented is simply $w_{a,x}$. However if x is latent and $m > 1$ we need to build a model which gives the probability of receiving a click on the set of arms A as well as determining which arm (if any) is clicked. As described earlier we assume that at most one arm is clicked at a given time so we cannot simply sum the rewards from individual arms.

3.2 Click Models

We describe a statistical model of which arm, if any, a user selects at each time. The models used come from Edwards & Leslie (2018) (building on work by El-Arini et al. 2009) but will be repeated here. The *click through rate* (CTR) will refer to the expected probability over all relevant unknowns (if any) of a click on some arm in a set of arms. The term *arm CTR* will be used if we are interested in the probability of a click on a specific arm.

A simple and popular model that addresses the question of *which* arm is clicked is the *cascade model* (Craswell et al. 2008). Arms are presented in order and the user considers each one in turn until one is clicked or there are no more left. An issue with this model is that the arm CTR is unaffected by position while, in reality, users would likely lose interest before looking at arms later in the list. However, it is not the purpose of this work to address these issues and the framework given here can readily be adapted to more complex models. For more on alternatives to the cascade model see Chuklin et al. (2015).

Using the cascade model we now give models to determine the CTR of a set of arms. We will work with two intuitive models which are motivated by the idea that there will be redundancy in sets that contain arms that are very similar to one another (if the user isn't interested in an arm then they are unlikely to be interested in any arm that is similar). The consequences of model choice will be discussed later in this section. Each model is initially specified for known x_t then extended to latent x_t at the end of this section.

Definition 3.1 (Probabilistic Click Model)**.** *In the Probabilistic Click Model (PCM) the user considers each arm in A_t independently in turn until they click one or run out of arms. At each step, the click probability for arm a is w_{a,x_t}. Therefore the CTR of the* *set A_t for known $\mathbf{w}_{\mathcal{A}}$ and x_t is*

$$r_{\text{PCM}}(x_t, A_t, \mathbf{w}_{A_t}) = 1 - \prod_{a \in A_t} (1 - w_{a,x_t}).$$

An issue with PCM is that a set of two identical arms gives a higher CTR than a single such arm. The next model avoids this unrealistic feature.

Definition 3.2 (Threshold Click Model)**.** *In the Threshold Click Model (TCM) each user has a threshold u_t drawn independently from distribution $U(0, 1)$. They consider each arm in turn, clicking the first arm $a \in A_t$ such that $w_{a,x_t} > u_t$. The CTR of the set is thus the probability that U is less than the maximal w_{a,x_t}:*

$$r_{\text{TCM}}(x_t, A_t, \mathbf{w}_{A_t}) = \max_{a \in A_t} w_{a,x_t}.$$

The TCM represents a user who, with state x_t, will click an advert if its relevance w_{a,x_t} exceeds a user-specific threshold u_t.

Since x_t is unobserved a more important quantity is the expected reward over $\mathbf{q}$. We write this, the CTR, for any click model π, as

$$\text{CTR}_\pi(A_t, \mathbf{q}_t, \mathbf{w}_{A_t}) = \mathbb{E}_{x_t \sim \mathbf{q}_t}[r_\pi(x_t, A_t, \mathbf{w}_{A_t})]. \tag{1}$$

This expectation is easy and fast to calculate as $\mathbf{q}$ defines a discrete distribution. Importantly, the latent x_t, means that in (1) arms are not independent for either click model.

The two click models differ in their effect on the *diversity* on optimal sets. Edwards & Leslie (2018) found that adverts in sets chosen to optimise CTR_{TCM} were less similar to each other than those chosen to optimise CTR_{PCM} and there was evidence that assuming PCM could lead to choosing sets with undesirable redundancy. Therefore we will develop and test both models in this work so that applications retain flexibility in choice of model. Edwards & Leslie (2018) also gave a continuum of models between PCM and TCM to provide intermediate set diversity and the methods and results given here hold for those click models.

3.3 Solution Method and Behaviour with Known Weights

Where arm weights are known the CTR, as given in Equation 1, is our objective function. Maximising the immediate reward (exploiting) is usually the simpler part of the bandit problems compared to calculating the value of exploration. However, maximising CTR is not straightforward as there is a combinatorial explosion in the number of available arm sets, online evaluation of which is computationally impractical for the intended web-based application.

The reward functions for both PCM and TCM were shown in Edwards & Leslie (2018) to possess a property, *submodularity*, for which there is a simple but effective heuristic algorithm. Submodularity in our context captures the intuitive idea of diminishing returns with advert set size - adding an advert to a large set gives a smaller increase in set CTR than adding it to a smaller subset.

Maximising a submodular function is NP-hard but for monotone submodular functions a computationally feasible greedy heuristic algorithm is known to have good properties (Nemhauser & Wolsey 1978). This algorithm starts with the empty set then selects arms iteratively, at each stage adding the arm that most increases the objective function (1). It will be referred to here as the **sequential algorithm (SEQ)** and is shown in Algorithm 1. Its calculation time scales linearly with km so it can be employed efficiently with large scale problems, unlike any method that tries to enumerate arm combinations.

Algorithm 1 Sequential Algorithm for Set Selection

```
Input: A set of available arms $\mathcal{A}$ with weights $\mathbf{w}_{\mathcal{A}}$; a click model $\pi$; a topic preference
  vector $\mathbf{q}_t$; a number of arms $m$ to select.
  Set $A = \emptyset$.
  for each slot $i$ in $1, 2, \ldots, m$ do
    Set $A_i = \arg\max_{a \in \mathcal{A}} \mathrm{CTR}_\pi(A \cup a, \mathbf{q}_t, \mathbf{w}_{A \cup a}) - \mathrm{CTR}_\pi(A, \mathbf{q}_t, \mathbf{w}_A)$.
    Set $\mathcal{A} \leftarrow \mathcal{A} \setminus A_i$.
  end for
Output: The set $A$ of arms to display.
```

Computational studies in Edwards & Leslie (2018) found that performance was very close to optimal when arm weights are known. In Section 6 we will test how well SEQ performs when arm weights are not known exactly. This will include the effect of click model misspecification and so we now describe a model which is not dependent on the click model, the **Naive algorithm (NAI)**. This selects the top m elements as ranked in order of independent element CTR $\mathbb{E}_{x \sim q_t} w_{a,x} = \mathbf{q}_t \cdot \mathbf{w}_a$.

4 Inference

The arm weights are initially unknown but can be learnt over time. This section will give an inference model for learning the weights from user actions. A model of learning for weights requires an estimate of our current knowledge of each weight (a point estimate and an estimate of uncertainty) together with a method to record how that knowledge changes as user actions are observed. User click behaviour depends on their state and the advert weights but, crucially, we do not observe the user state x so we do not know which weight to attribute click behaviour to. We address these issues by using a Bayesian framework to incorporate knowledge of each user's topic preference vector $\mathbf{q}$. The Bayesian model is presented in Section 4.1. There are practical computational problems with implementing this exactly so an approximate version is detailed in Section 4.2. The method is analysed in Section 4.3 together with a discussion of the conditions on $\mathbf{q}$ required for learning to be reliable.

4.1 Feedback and Learning

We will quantify our knowledge of weights with a joint probability distribution over all arm weights with density given by $p(\mathbf{w}_{\mathcal{A}})$. A single step of learning then proceeds as follows. We are presented with a user with topic preference $\mathbf{q}$. In response we choose a set of arms A, to which the user gives feedback in the form of either a click (together with which arm was clicked) or no click. Based on this feedback, the belief density $p(\mathbf{w}_{\mathcal{A}})$ is updated which is then used in the next step. Since only a single updating step is described in this section no time subscripts will be used in the notation for simplicity. Updating through time is simply a series of single updating steps. Formally $p(\mathbf{w}_{\mathcal{A}})$ would then need to be conditioned on the history of all relevant information up to the current time but, again, for notational simplicity this is omitted here.

User feedback is summarised with two new variables: y where $y = 1$ if the user clicked some arm and $y = 0$ otherwise, and m^* the number of arms considered by the user. Under the cascade model $m^* = i$ if arm a_i is clicked or $m^* = m$ if no arm is clicked. This distinguishes between two possible interpretations for arms that are not clicked. Arms $a_i, i \leq m^*$ are considered by the user so we receive information which affects the updating, but arms $a_i, i > m^*$ are not considered by the user so no information is received. To simplify notation in the following it is useful to define the set of arms considered by the user but not clicked as

$$A' = \begin{cases} A & \text{if } y = 0 \\ \{a_1, \ldots, a_{m^*-1}\} & \text{if } y = 1. \end{cases}$$

The full updating equations for both PCM and TCM are given in Appendix A.1. These involve finding the joint distribution over a large number of variables which cannot be decomposed due to dependency on $\mathbf{q}$. Not knowing the state x means we do not know which $w_{a,x}$ to attribute any click or refusal to click. TCM has the added complication of dependency on the latent user threshold u which is common to all arms. This means conjugate updates are not possible and exact updating is impractical. The next section will describe an alternative approximate updating method.

4.2 Updating Weight Beliefs

The standard way to resolve the issues in updating caused by dependency on latent variables, such as found in the previous section, is to use an *expectation maximisation* algorithm for which online versions exist (e.g. Cappé & Moulines 2009, Larsen et al. 2010). The approach used is to sample an $\tilde{x}$ from the belief distribution for x and then update the weights using $\tilde{x}$ as the state. By conditioning on $\tilde{x}$ instead of $\mathbf{q}$ we can treat user actions for any arm a as a Bernoulli trial and attribute successes or failures to $w_{a,\tilde{x}}$. For PCM this allows beliefs for all points to be independent Beta distributions. Each weight $w_{a,x}$ has a belief distribution $W_{a,x} \sim Beta(\alpha_{a,x}, \beta_{a,x})$ and the joint distribution of the weight beliefs for all arms in $\mathcal{A}$ is $\mathbf{W}_{\mathcal{A}}$. The belief state is given by the α and β values so $2kn$ values are required to store the belief state. The model is now conjugate and the update conditional on $\tilde{x}$ is given by $\alpha_{a_{m^*},\tilde{x}} \leftarrow \alpha_{a_{m^*},\tilde{x}} + y$ and $\beta_{a,\tilde{x}} \leftarrow \beta_{a,\tilde{x}} + 1$ for all $a \in A'$ with all other α, β values unchanged.

A key observation in Larsen et al. (2010) is that the belief distribution from which $\tilde{x}$ is drawn should be dependent on the user feedback just observed (rather than just $\mathbf{q}$). This posterior $\tilde{\mathbf{q}} = (\tilde{q}_1, \ldots, \tilde{q}_n)$ depends on $\mathbf{W}_{\mathcal{A}}$, y, m^*, $\mathbf{q}$ and A. The detail of the derivation of $\tilde{\mathbf{q}}$ is given in Appendix A.2. Under PCM and conditioning on x we are able to obtain an easy to calculate formula for $\tilde{\mathbf{q}}$:

$$\tilde{q}_x = q_x \frac{(\mu_{a_{m^*},x})^y \prod_{a \in A'} (1 - \mu_{a,x})}{\sum_{j=1}^{n} [q_j (\mu_{a_{m^*},x})^y \prod_{a \in A'} (1 - \mu_{a,x})]} \,. \tag{2}$$

The complete method for updating is shown in Algorithm 2.

Algorithm 2 Posterior Sampled Bayesian Updating

```
Input: A set of arms $A$ presented to the user; weight belief distributions $\mathbf{W}_A$ parameterised by $\{\alpha_{a,i}, \beta_{a,i} \mid a \in A, i = 1, \ldots, n\}$; a user response given by $y$ and $m^*$; the state probability vector $\mathbf{q}$ for the $n$ states.
    Calculate the posterior state probabilities $\tilde{\mathbf{q}} = (\tilde{q}_1, \ldots, \tilde{q}_n)$, given in (2).
    Draw $\tilde{x}$ from $\tilde{\mathbf{q}}$.
    Update weight beliefs:
        $\alpha_{a_{m^*},\tilde{x}} \leftarrow \alpha_{a_{m^*},\tilde{x}} + y$,
        $\beta_{a,\tilde{x}} \leftarrow \beta_{a,\tilde{x}} + 1$, for all $a \in A'$,
        and all other $\alpha_{a,x}, \beta_{a,x}$ unchanged.
Output: A set of updated belief distributions $\mathbf{W}_A$.
```

By studying the stochastic approximation methods (e.g. Larsen et al. 2010) it becomes clear that deterministic averaging over x is equally valid. This is done by updating elements of $\mathbf{W}_{\mathcal{A}}$ in proportion to $\tilde{\mathbf{q}}$, that is, by setting

$$\begin{aligned} &\alpha_{a_{m^*},x} \leftarrow \alpha_{a_{m^*},x} + y\tilde{q}_x, \\ &\beta_{a,x} \leftarrow \beta_{a,x} + \tilde{q}_x \text{ for all } a \in A' \end{aligned} \tag{3}$$

All other α, β values are unchanged as before. These two methods will be compared in Section 4.3.

For TCM, click probabilities are not independent even when x is known and therefore it does not reduce to a simple updating model even given a sampled x. The updating for known x for TCM (given in Appendix A.3) does not fit into a simple updating scheme. In addition, to use this in the posterior sampled Bayesian updating in Algorithm 2 requires taking integrals over the multiple belief distributions to find the posterior for $\mathbf{q}$. Therefore the heuristic updating method used for PCM will not work for TCM. The approach we use to handle this difficulty is to record and update beliefs as though the click model is PCM. For the arm a_1 in the first slot the models are the same but for subsequent arms we are making an independence assumption. This does not utilise all available information but it will be shown empirically in simulations in Section 6 that weight beliefs converge to the true weights as effectively under this method as when the PCM is the true click model.

4.3 Approximate Updating Analysis

In Section 4.2 two approximate updating methods were given, each using $\tilde{\mathbf{q}}$, the posterior of $\mathbf{q}$ given user actions. Algorithm 2 updates using a sampled value from $\tilde{\mathbf{q}}$ while (3) updates deterministically in proportion to $\tilde{\mathbf{q}}$. These can be compared by using the single arm case where $k = m = 1$ so that the results are unaffected by click model or set choosing algorithms.

The simulation has 500 runs, each with $N = 1000$ time steps. In each run, for each time t, a $\mathbf{q}_t$ is drawn i.i.d. from a Dirichlet distribution. There are n possible states for x so each Dirichlet distribution has n parameters which here are set to $1/n$. At each time a user click is simulated for the single arm set, beliefs are updated, then the absolute error $|\mu_{1,x} - w_{1,x}|$ between the current mean belief $\mu_{1,x}$ and the true weight $w_{1,x}$ for each state x is recorded. This error, averaged over all the runs and x, is shown on the left in Figure 1. In addition, the state thought to have the highest weight $\text{argmax}_x(\mu_{1,x})$ was compared to the truth $\text{argmax}_x(w_{1,x})$. The proportion that this was incorrect is shown on the right in Figure 1. This last metric tests an issue found to happen occasionally in a similar problem in Larsen et al. (2010) where the updating mixes up the state weights. On both measures the deterministic version performed better. Similar patterns are found with other values of n and β.

Note that accurate inference relies on $\mathbf{q}_t$ being varied over time since if $\mathbf{q}_t$ is fixed then inference is unreliable as illustrated in Figure 2. The weight posterior distributions are still converging but often to the wrong value. This is an identifiability issue due to there being insufficient information to solve the problem, rather than an issue with the updating method. This can be seen by considering the offline version of the problem for the simplest case where $n = 2$ given T observations. We then have a system of equations $\mathbf{y} = Q\mathbf{w} + \boldsymbol{\epsilon}$ where $\mathbf{y}$ is a vector of T observed rewards, Q is a $T \times 2$ matrix where each row t is $\mathbf{q}_t$, and $\boldsymbol{\epsilon}$ is a noise term. The least squares solution is given by $(Q^\top Q)^{-1}Q^\top \mathbf{y}$ which has a unique solution if and only if Q is of rank 2. Therefore using a constant $\mathbf{q}_t$ will not give a single solution.

This may be important in practice since it indicates that we cannot reliably learn the qualities of adverts by observing clicks with only a single search term or very similar search terms. Indeed, simulations suggest that the algorithm attributes rewards most accurately and reliably when each q_x is sometimes large which motivates the use of feedback for a variety of search terms. This has implications for Hauser et al. (2009) where prior beliefs for user cognitive styles are generated from a priming study and so are the same for all users. This is equivalent to using a constant $\mathbf{q}$ over time which we have shown to be unreliable.

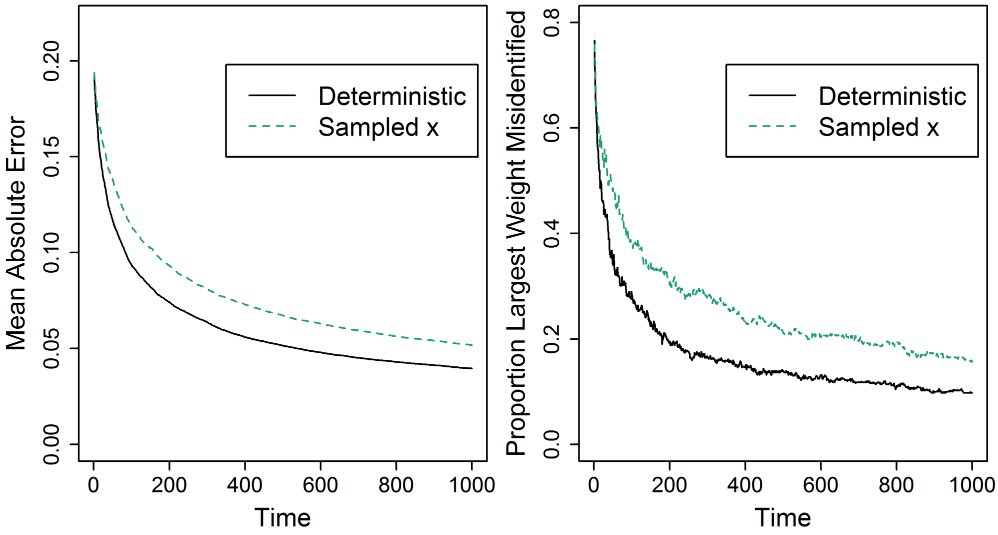

Figure 1: The mean absolute error of $\hat{w}$ (left) and the proportion of times that the highest weight is misidentified (right) over 500 runs for a single arm with $n = 5$, $\alpha = 1$ and $\beta = 2$.

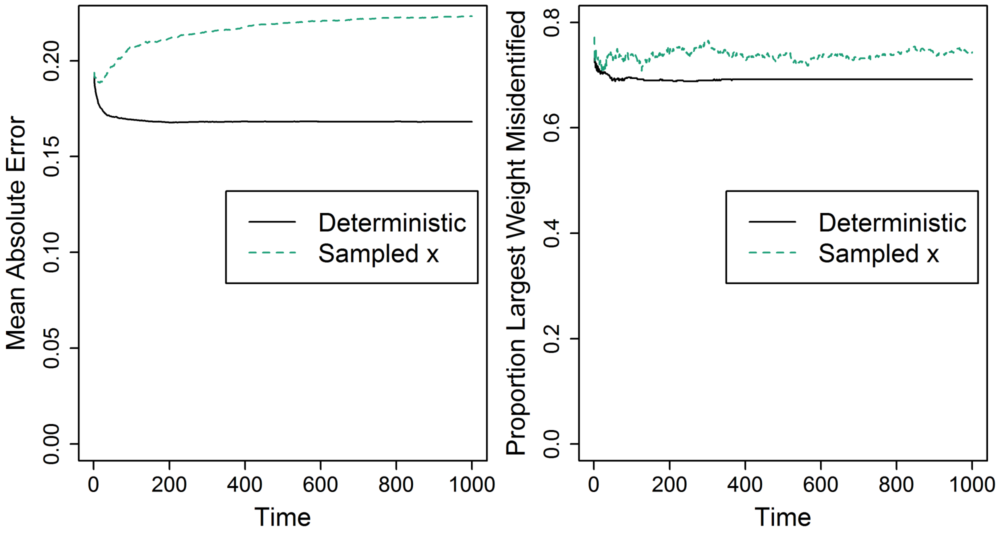

Figure 2: The mean absolute error of $\hat{w}$ (left) and the proportion of times that the highest weight is misidentified (right) over 500 runs for a single arm with $n = 5$, $\alpha = 1$ and $\beta = 2$ and each $\mathbf{q}_t = (0.8, 0.05, 0.05, 0.05, 0.05)$.

5 Exploring the Weights

Section 3.3 gave the SEQ algorithm for selecting a set of arms when weights are known. This corresponds to the exploitation part of a bandit problem. With weights initially unknown and learned from experience it is also necessary to explore by choosing arms to learn the true weights effectively without neglecting immediate reward. The objective is still to achieve high CTR but over all future times as well as the current time. The exact objective and time frame will depend on the application and business objectives. In this section a method is given whereby a range of existing bandit algorithms (or *policies*) can be easily adapted to learn the true weights while still retaining the good short term CTR performance of the SEQ algorithm as weights become known.

Index policies are a family of bandit policies which assign to each arm a real valued index and then chooses the arm with the highest index. Examples include: the *Greedy policy* for which the index is the expected immediate reward; the *Gittins index* (Gittins et al. 2011) where the index is found by solving a stochastic optimisation problem on each arm; *Thompson sampling* (e.g. May et al. 2012) which uses a stochastic index by sampling from the posterior for the arm; and the *upper-confidence-bound* policy family (e.g. Auer et al. 2002) where the index is based on an optimistic upper bound of the arm's value.

To adapt these methods for the multiple advert problem the index is substituted for the true weight in SEQ (or other set selection algorithm). Let I_t be the history up to time t which consists of all available information that is relevant to the current decision, namely $\mathbf{q}_1, \ldots, \mathbf{q}_{t-1}$; the weight priors; past actions $A_1, \ldots, A_{t-1}$; and user responses given by $m_1^*, \ldots, m_{t-1}^*$ and $y_1, \ldots, y_{t-1}$. For the purposes of choosing arms this information is used only via the current posterior weight belief distributions $\mathbf{W}_{\mathcal{A},t}$. Then, formally, a policy for our problem consists of two parts: (1) an exploration algorithm $\nu(\mathbf{W}_{\mathcal{A},t}|I_t)$ which maps $\mathbf{W}_{\mathcal{A},t}$ to real valued indices $\tilde{\mathbf{w}}_{\mathcal{A},t}$, and (2) a set selection algorithm $S(\mathcal{A}, \tilde{\mathbf{w}}_{\mathcal{A},t}, m)$ which takes $\tilde{\mathbf{w}}_{\mathcal{A},t}$ and outputs a set A of m chosen arms. Each $\tilde{w}_{a,t,i}$ in $\tilde{\mathbf{w}}_{\mathcal{A},t}$ can be thought of some proxy for the unknown weight $w_{a,t,i}$.

We will not attempt to compare the general performance of the many available index policies since these have been widely studied on simpler bandit problems. Instead we will concentrate on adapting and testing one, Thompson sampling. Thompson sampling is chosen because it fits the requirements of this problem, namely that it is fast to compute and in similar long horizon bandit problems it has been shown to work well and explore effectively (see e.g. Russo & Van Roy 2014, May et al. 2012). The method is given in Algorithm 3.

In order to learn the best arm sets for any user it is necessary for the algorithm to sample infinitely often from all arms (so that it never stops learning completely). Theorem 5.1 below gives conditions under which Algorithm 3 using the deterministic updating scheme from Section 4.2 will select each arm infinitely often. The strongest condition is on the distribution of each $\mathbf{q}_t$, and will be discussed after the theorem.

Theorem 5.1. *Let $q^* = \inf_{t,x} \Pr(q_{t,x} = 1) > 0$. Then the multiple action Thompson sampling algorithm given in Algorithm 3 with SEQ as set choosing method sam-*

Algorithm 3 Multiple Action Thompson Sampling

```
Input: The available arms $\mathcal{A}$ with posterior weight beliefs $\mathbf{W}_{\mathcal{A},t}$ where each $W_{a,t,i}$ is a $Beta(\alpha_{a,t,i}, \beta_{a,t,i})$ distribution; the number of arms to be selected $m$; a set choosing algorithm $S(\mathcal{A}, \tilde{\mathbf{w}}_{\mathcal{A},t}, m)$.
  for all $a \in \mathcal{A}, i = 1, \ldots, n$ do
    Draw $\tilde{w}_{a,t,i} \sim W_{a,t,i}$
  end for
  Select an arm set using set choosing algorithm $S(\mathcal{A}, \tilde{\mathbf{w}}_{\mathcal{A},t}, m)$
Output: A set of chosen arms $A$ of size $m$.
```

ples infinitely often from each arm for any click model from Section 3.2. That is, $\Pr(|\tau_{a,T}| \to \infty$ *as* $T \to \infty) = 1$ *for any arm* $a \in \mathcal{A}$*, where* $\tau_{a,T}$ *is set of times* $t = 1, .., T$ *that* $a \in A_t$*.*

Proof. See Appendix C □

The condition that all $q^* > 0$ in Theorem 5.1 is unlikely to hold in practice. It is needed for our proof method to eliminate the case where $\mu_{a,t,x} \to 1$ as $|\tau_{a,t}| \to \infty$ even though $w_{a,x} < 1$. However, the conditions for this to happen are extremely unlikely to occur as the prior $W_{a,0,x}$ would have to be concentrated close to 1. In practice, as found in Larsen et al. (2010) and discussed in Section 4 it is sufficient for each q_x to be sometimes large.

To summarise, this work is the first to formulate a contextual bandit problem where the context takes the form of a discrete probability distribution. This form of context represents uncertainty about a latent state which creates challenges in inference for the arm weights. In Section 4 we gave a computationally fast approximate Bayesian method for inference and learning which overcomes these problems. In this section we gave a new method by which many existing bandit algorithms can be adapted to work with submodular set choosing methods and illustrate this method with a policy using Thompson sampling. The next section will test all the parts of our solution method as a complete policy in simulations of the full multiple adverts problem.

6 Computational Experiments

This section uses simulations to test the complete solution method proposed in Section 5 and given here in Algorithm 4.

6.1 Regret Simulations

The performance measure used is the *cumulative regret* over time. This compares the expected reward of policies to the *SORACLE* policy which uses SEQ to select arms with knowledge of the true weights and assuming the true click model. At any time T

Algorithm 4 Full Multiple Adverts Algorithm

```
Input: The available arms $\mathcal{A}$ with prior weight beliefs $\mathbf{W}_{\mathcal{A},0}$; the number of arms to be
  selected $m$; a multiple action policy $\psi$; a click model $\pi$.
  for $t = 1, \dots, T$ do
    Select a set of $m$ arms using Algorithm 3.
    The user responds according to click model $\pi$.
    Update arm weight beliefs using (3).
  end for
Output: $T$ chosen arm sets with a user click history.
```

the cumulative regret is

$$\frac{1}{T}\sum_{t=1}^{T}\left[CTR_{\pi}(A_t^{SORACLE}, \mathbf{w}_{\mathcal{A}}, \mathbf{q}_t) - CTR_{\pi}(A_t^{\psi}, \mathbf{w}_{\mathcal{A}}, \mathbf{q}_t)\right]$$

where CTR$_\pi$ is the CTR with click model π, and A_t^{ψ} and $A_t^{SORACLE}$ are the arm sets chosen by, respectively, the policy ψ being tested and SORACLE.

For each simulation run a policy is selected. This consists of a set choosing method (either SEQ or NAI) and a bandit exploration algorithm (either Thompson sampling (TS) or the Greedy policy, which uses the posterior mean as an estimate of the true weights). Each policy is denoted by a two-part name where the first part gives the set choosing algorithm, either N for NAI or S for SEQ, and the second part gives the bandit algorithm, either TS or G for Greedy.

Two sets of experiments are reported in this section. The first uses randomly generated arm weights while the second uses a scenario with fixed arm weights with low expected CTR.

6.1.1 Set 1

The true weights for the run are independently drawn from a mixture distribution where each is *relevant* with probability $\xi = 0.5$ and *non-relevant* otherwise. If relevant the weight is drawn from a $Beta(\alpha = 1, \beta = 2)$ distribution, otherwise the weight is 0.001. Each weight is given an independent $Beta(1, 2)$ prior belief (all assumed a priori to be relevant). At each time $t = 1, 2, \dots, T$ a state distribution $\mathbf{q}_t$ is sampled from a Dirichlet distribution with all n parameters equal to $1/n$. Note that this does not satisfy the assumption on $\mathbf{q}$ given in Theorem 5.1. In response the policy chooses a set of m arms from the available k. A user action is then simulated using π, and weight beliefs updated. The values of $k = |\mathcal{A}|$, $m = |A_t|$, T and n used are given with the results. This is repeated for all policies using the same weights, $\mathbf{q}_t$ and common random numbers. Both PCM and TCM are used as true click models to generate the rewards which are given to the algorithm. The SEQ-based policies will use the correct click model which will be appended to its name e.g. STS-PCM. Section 6.2 will consider learning when the click model is misspecified.

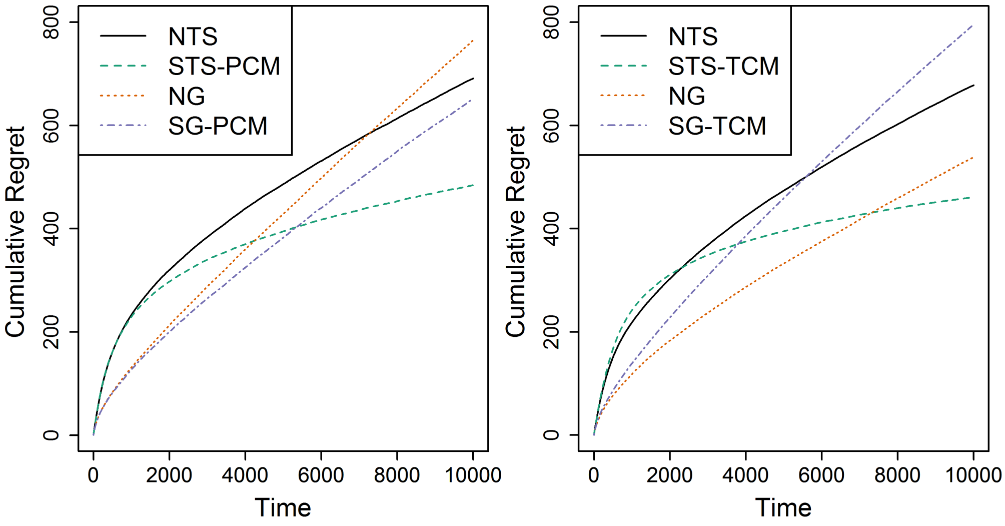

Figure 3: Mean cumulative sequential regret. Simulation setting are $n = 5$, $k = 20$, $m = 2$, $\xi = 0.5$, $\alpha = 1$ and $\beta = 2$. The true click model is PCM on the left and TCM on the right.

At each time, for each of the 500 simulation runs and each policy the cumulative regret is calculated using the chosen arm set and the true weights.

Two different sizes of problem are used to give an idea of how the learning rates scale. The cumulative regret averaged over all runs is shown in Figures 3 and 4 for the smaller and larger problems respectively. The overall pattern is the same for each but on different timescales.

Some aspects of the results are as would be expected. For policies using TS, as the weights are learnt, SEQ chooses higher reward sets than NAI and so STS outperforms NTS. Greedy is more effective than TS at earlier times but does not learn well and so falls behind later. It takes longer for the superior learning of TS to pay off for TCM than PCM.

There are some surprises though. The two Greedy policies perform very differently on PCM and TCM. SG does better than NG on PCM as would be expected but on TCM the order is reversed indicating that the SG-TCM does not learn well. It appears that NTS does not learn well as it is slow to catch up with the Greedy policies (and may not be catching the best of the Greedy policies at all on the larger problem). This supports the use of combination of SEQ and TS on this problem. Further simulations in Section 6.2 will look at learning rates and these issues in more detail.

6.1.2 Set 2

The experiments in the second set are setup as in the first set except that the arm weights are fixed over all runs. This allows us to observe policy behaviour in more detail which

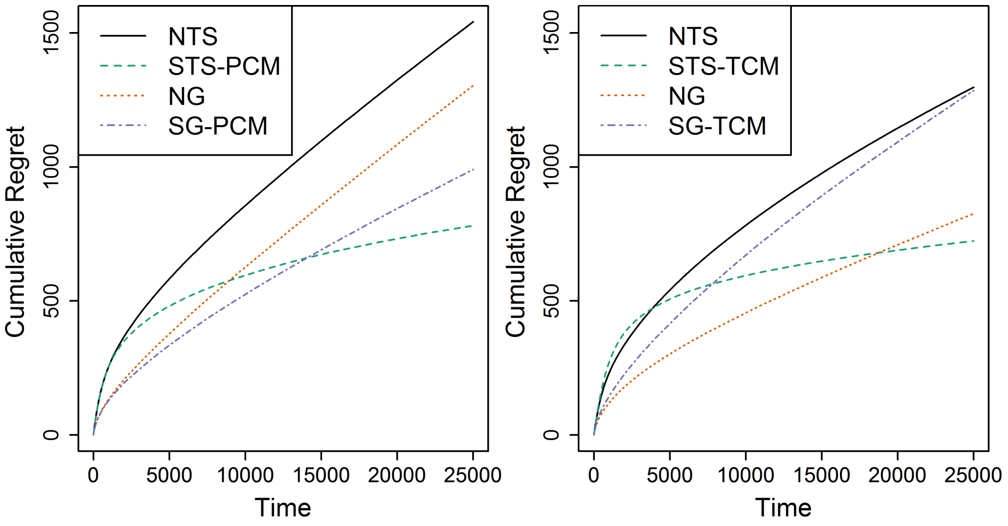

Figure 4: Mean cumulative sequential regret. Simulation setting are $n = 10$, $k = 40$, $m = 3$, $\xi = 0.5$, $\alpha = 1$ and $\beta = 2$. The true click model is PCM on the left and TCM on the right.

State	Arm number									
	1	2	3	4	5	6	7	8	9	10
1	0.03	0	0	0.015	0.015	0	0.01	0.021	0	0
2	0	0.03	0	0.015	0	0.015	0.01	0	0.021	0
3	0	0	0.03	0	0.015	0.015	0.01	0	0	0.021

Table 1: The arm weights for experiment set 2. Arms 1-3 are type A, 4-7 are type B, and 8-10 are type C.

highlights differences between problems using PCM as true click models and those using TCM. In these experiments we select arm weights to be much smaller than in the first set, creating a more challenging and realistic scenario where the policies must distinguish between arms with similar but low CTRs.

We take $m = 2$ and $n = 3$, with $k = 10$ arms split into three types. Arms of type A and B all have the same mean weight 0.01. For the three arms of type A this is concentrated on a single state (a different state for each arm), while for four arms of type B the weight is split evenly between two or three states (a different combination of states for each arm). The three arms of type C have concentrated weight like type A but at 70% of the strength of type A. The arm weights are shown in Table 1. For TCM it is optimal to always choose two arms of type A but for PCM the correct action is more varied depending on $\mathbf{q}_t$.

The regret performance for each policy over $N = 40000$ time steps is shown in Figure 5. A longer time horizon is used because learning is slower with the smaller number of

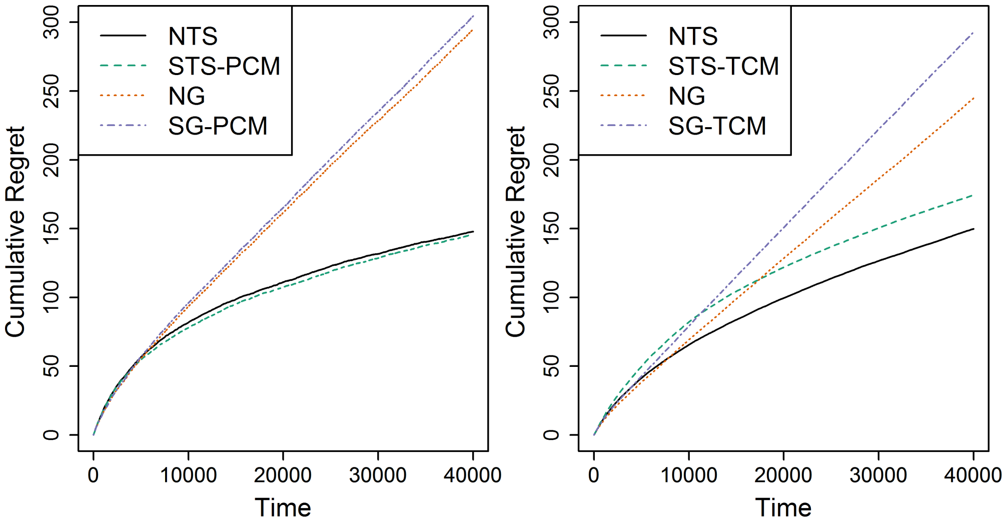

Figure 5: Mean cumulative sequential regret for experiment set 2. A fixed arm scenario with small weights is used. The true click model is PCM on the left and TCM on the right.

clicks that come with low CTRs. For all click models the two greedy policies struggle. With PCM the two TS-based policies are very similar. This is unsurprising since with low expected CTRs the reward function under PCM is a similar function to additive rewards and in this setting SEQ and NAI act similarly. For TCM the results are more unexpected with NTS outperforming STS, although with a similar pattern.

Figure 6 shows the arm types chosen by each policy at times $t \in \{35001, \ldots, 40000\}$. In the TCM problem it is optimal to choose arms of type A but the other policies choose arms from both of the other groups. The STS-TCM policy has chosen a higher percentage of optimal arms than NTS, showing that it doing better by the end of the simulation but, because the available reward is small, its cumulative regret performance is only catching slowly. In the PCM problem the optimal mix of arms consists of arms from all groups. Here the TS policies have a similar mix of types to the Oracle policy while the greedy policies select from type B (the all-rounders) too often, indicating a lack of adequate learning.

Overall, it appears from the experiments that both exploration algorithm and set choosing method are important for good performance. Which is more important varies between experiment, especially for short term performance. However, over the longer term, while the naive set choosing method can do quite well on some problems, using a greedy exploration algorithm consistently results in under-performance.

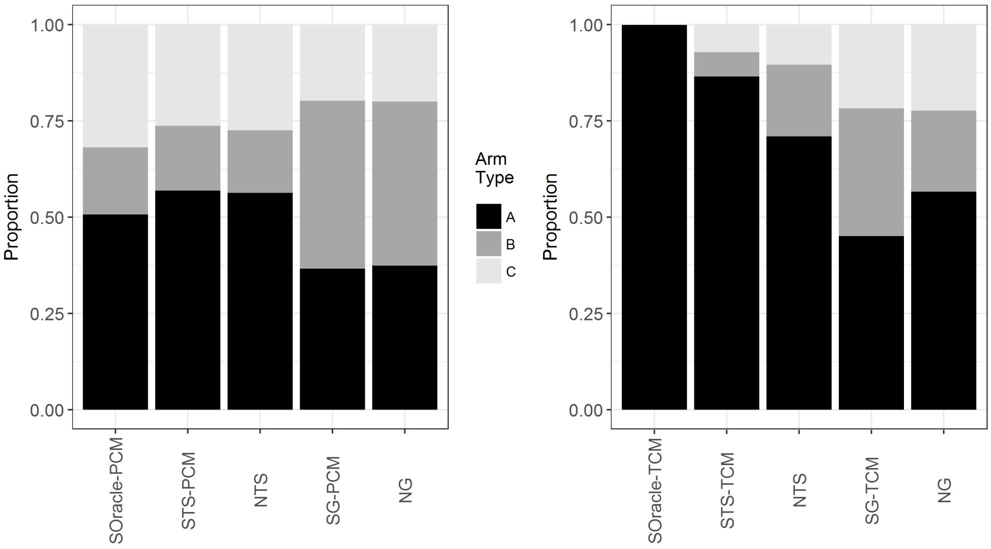

Figure 6: The actions for each policy in the last 5000 time steps in experiment set 2. The true click model is PCM on the left and TCM on the right.

6.2 State of Knowledge Simulations

Cumulative regret measures the overall performance of a policy but is not ideal for comparing how SEQ and NAI affect exploration as this can be obscured by the inferior set choosing ability of NAI for known weights. We give a new performance measure which separates the learning capabilities of an algorithm from the immediate CTR performance of its set choosing component. This allows us to directly examine how SEQ affects exploration and how this changes with the click model assumed by the policy (both correct and misspecified).

For Set 2 in Section 6.1 we could assess how well the algorithms had learnt by directly observing their selections towards the end of the simulation (Figure 6). To measure how effectively a policy has learnt in experiment set 1 of Section 6.1 we rerun these simulations but rather than record the cumulative regret, we instead use a new measure, the *greedy posterior regret* (GPR). To define this we first define the *greedy posterior reward* of a policy. At any time this is the expected reward using the SG policy with the true click model if no further learning occurs beyond that time. The GPR is then the difference between this value and the expected reward of the SORACLE policy which knows the true weights. This is a more useful measure of learning than more general measures such as the Kullback-Leibler difference because GPR gives a measure of *effective* learning by estimating how well the policy has focused on the arms likely to have larger weights which are therefore more likely to be part of optimal sets.

The simulation estimates the GPR for a policy at any time as follows. A $\mathbf{q}_t$ is generated as usual and a single action taken using the SG policy using the posterior mean of all weights at time t. This is repeated for 100 different simulated values of $\mathbf{q}_t$

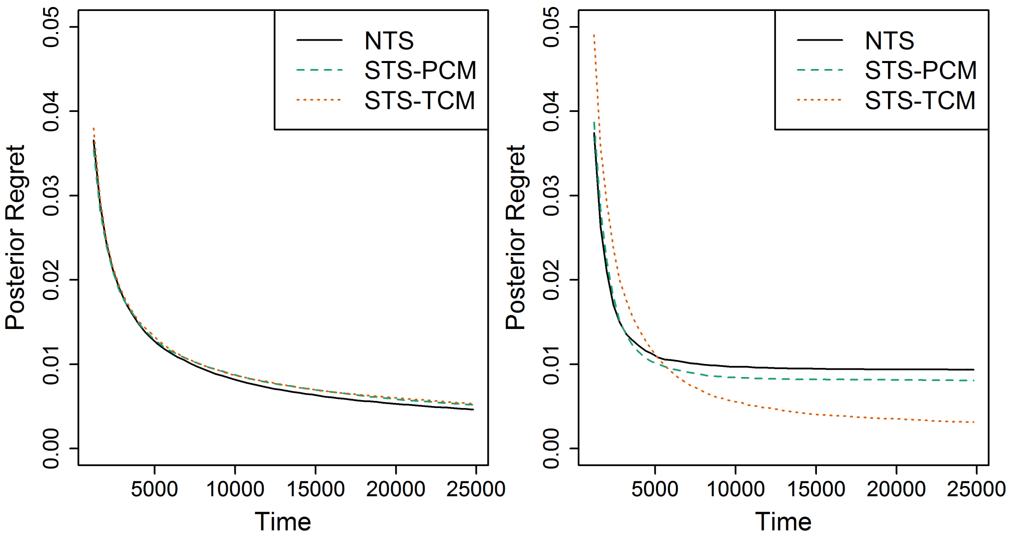

Figure 7: Mean greedy posterior regret for TS-based policies with PCM (left) and TCM (right). Simulation setting are $n = 10$, $k = 40$, $m = 3$, $\xi = 0.5$, $\alpha = 1$ and $\beta = 2$.

for each run of the original simulation. The GPR at time t is the mean regret over the 100 $\mathbf{q}_t$ values.

The GPR over time for TS-based policies with PCM and TCM as the true click model are shown in Figure 7. It also tests misspecified click models for SEQ-based policies. NTS ignores click model. The time axis starts at $t = 1000$ because GPR values are high at early times before the policies have had much time to learn. GPR values for the Greedy-based policies are much higher and so these are shown on a separate plot in Figure 8 with just NTS for comparison. We will consider the TS-based policies first before moving on to the Greedy-based policies. For PCM all of the TS policies are very similar. On TCM both STS policies do better than NTS with STS-TCM best of all. Generally, all the TS-based policies appear robust to variation in the click model.

The learning for the Greedy-based policies is, as expected, clearly worse than for TS but, in addition, there is large variation among the Greedy variants. In particular SG-TCM learns poorly on both PCM and TCM. This explains the poor regret for SG with TCM in Section 6.1, showing that the problem is with the click model assumed by the policy rather than the true click model. So, for Greedy-based policies, it is more robust for learning to assume PCM.

The results here and the regret simulations in Section 6.1 suggest that, in addition to superior exploitation, STS also learns as effectively as NTS and is better for a correctly specified TCM.

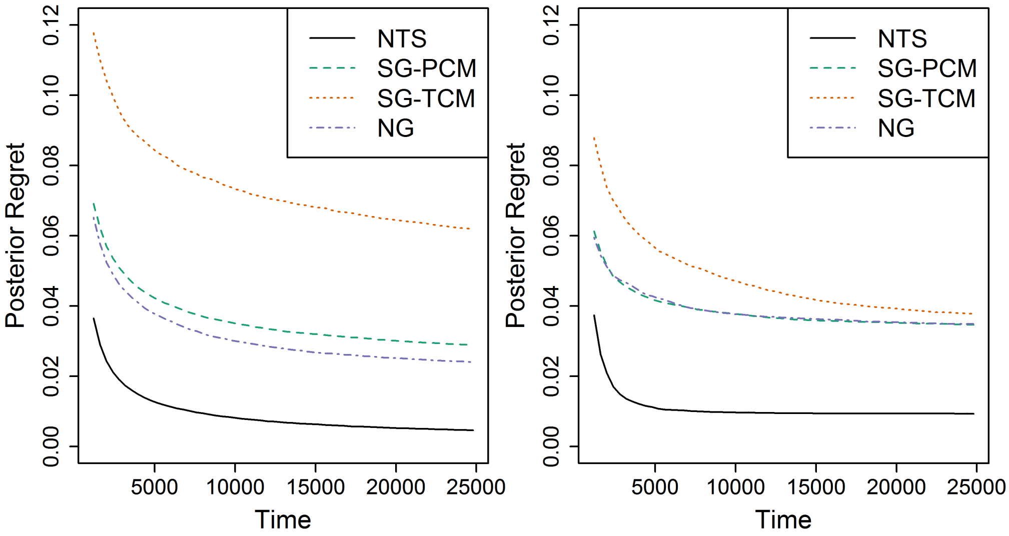

Figure 8: Mean greedy posterior regret for Greedy-based policies with PCM (left) and TCM (right). Simulation setting are $n = 10$, $k = 40$, $m = 3$, $\xi = 0.5$, $\alpha = 1$ and $\beta = 2$. The NTS policy is shown for comparison.

7 Discussion

In this paper we formulated the multiple advert problem and gave a solution method which combined bandit algorithms with submodular set-choosing methods in a Bayesian setting. The setting is a challenging one, principally due to the twin uncertainties concerning: (i) the topic and quality of each advert and, (ii) the preferences of the users. The first of these can be learnt over time by observing user click behaviour but the second uncertainty is inherent due to an unobservable user state. The resulting contextual bandit problem, our analysis and proposed methods will have wider application in problems where there is state uncertainty in addition to the usual arm uncertainty.

The model used here is structurally simple but can encompass greater complexity in application through the topic preference vector $\mathbf{q}$ which could contain considerable information since model and solution methods are practical with very large $\mathbf{q}$. A possible limitation of the model is that a user state is given by a single x. An easily implemented extension would be to record the user state as a very sparse vector the same length as $\mathbf{w}$ and model clicks with some function of these two vectors e.g. the probit of the dot product. A similar model for single advert CTR prediction that has been implemented in commercial web search is given in Graepel et al. (2010).

We did not compare our approach with existing methods because none were designed for our problem formulation and make inappropriate assumptions for our specific model. A formal bound on finite time regret would be desirable but the difficulty of our problem with a stochastic latent state in addition to stochastic arms makes this impractical. In particular the approximate Bayesian scheme does not give guarantees of accurate convergence of weight beliefs so regret growth cannot be usefully bounded.

The simulations in Section 6.1 suggest exploration of weights will be slow as the problem is scaled up. This is mainly due to there being kn parameters to learn compared to just k for the standard MAB but also because the unknown x makes feedback less informative than normal. Therefore slow exploration is a feature of the problem rather than the methods used and so is unavoidable unless greater assumptions (e.g. dependence between weights or arms) are made on the structure of the weights. There are examples of this in bandit problems (e.g. Yue & Guestrin 2011) but the intention in this work is to use a model that is as general as possible, only adding in such assumptions if it is clear they are valid and necessary. Furthermore we anticipate that exploration may not be a problem in practice due to the high rate of observations and by using priors that represent existing knowledge of likely arm CTRs. This ability to incorporate prior knowledge is a strong advantage of the proposed Bayesian scheme.

8 Acknowledgements

This work was funded by a Google Faculty Research Award. James Edwards was supported by the EPSRC funded EP/H023151/1 STOR-i CDT.

A Derivations for Section 4

A.1 Updating Equations for Arm Weights

PCM. The joint distribution for all weights given a feedback step is updated as given below. In the following note that $p(\cdot|\mathbf{q},x)$ simplifies to $p(\cdot|x)$ and that $p(\mathbf{w}_\mathcal{A},x|\mathbf{q},A) = p(\mathbf{w}_\mathcal{A})q_x$ since x is independent of $\mathbf{w}_\mathcal{A}$. The posterior belief for the weights after a user action is,

$$\begin{aligned}
p(\mathbf{w}_\mathcal{A}|y,m^*,\mathbf{q},A) &= \sum_{x=1}^{n} p(\mathbf{w}_\mathcal{A},x|y,m^*,\mathbf{q},A)\\
&= \sum_{x=1}^{n} \frac{p(y,m^*|\mathbf{w}_\mathcal{A},x,\mathbf{q},A)p(\mathbf{w}_\mathcal{A},x|\mathbf{q},A)}{p(y,m^*|\mathbf{q},A)}\\
&= \frac{1}{p(y,m^*|\mathbf{q},A)}\sum_{x=1}^{n}\left\{\left[(w_{a_{m^*},x})^y\prod_{a\in A'}(1-w_{a,x})\right]p(\mathbf{w}_\mathcal{A})q_x\right\}\\
&= \frac{p(\mathbf{w}_\mathcal{A})}{p(y,m^*|\mathbf{q},A)}\sum_{x=1}^{n}\left[q_x(w_{a_{m^*},x})^y\prod_{a\in A'}(1-w_{a,x})\right].
\end{aligned}$$

TCM. The updating equation for $\mathbf{w}_\mathcal{A}$ for TCM is similar to that for PCM except that $p(y, m^*|\mathbf{w}_{\mathcal{A}}, x, \mathbf{q})$ changes due to the user threshold u:

$$\begin{aligned}
p(\mathbf{w}_{\mathcal{A}}|y, m^*, \mathbf{q}, A) &= \sum_{x=1}^{n} p(\mathbf{w}_{\mathcal{A}}, x|y, m^*, \mathbf{q}, A) \\
&= \sum_{x=1}^{n} \frac{p(y, m^*|\mathbf{w}_{\mathcal{A}}, x, \mathbf{q}, A) p(\mathbf{w}_{\mathcal{A}}, x|\mathbf{q}, A)}{p(y, m^*|\mathbf{q}, A)} \\
&= \frac{p(\mathbf{w}_{\mathcal{A}})}{p(y, m^*|\mathbf{q}, A)} \int_{u=0}^{1} \sum_{x=1}^{n} \left[q_x (\mathbb{1}_{\{w_{a_{m^*}, x} > u\}})^y \prod_{a \in A'} \mathbb{1}_{\{w_{a,x} \leq u\}} \right] du. \qquad (4)
\end{aligned}$$

A.2 Derivation of $\tilde{\mathbf{q}}$

The posterior $\tilde{\mathbf{q}} = (\tilde{q}_1, \ldots, \tilde{q}_n)$ depends on $\mathbf{W}_{\mathcal{A}}$, y, m^*, $\mathbf{q}$ and A. For ease of reading the rest of this section will use $\mathbf{w}$ and $\mathbf{W}$ to respectively stand for $\mathbf{w}_{\mathcal{A}}$ and $\mathbf{W}_{\mathcal{A}}$. Bayes Theorem will be used to condition the outcome on x which allows the use of the conditional independence of arms under PCM to factorise to a simple formula.

$$\begin{aligned}
\tilde{q}_x &= p(x \mid \mathbf{W}, y, m^*, \mathbf{q}, A) \\
&= \int p(x, \mathbf{w} \mid \mathbf{W}, y, m^*, a, A) \, \mathrm{d}\mathbf{w} \\
&= \int p(x \mid \mathbf{w}, y, m^*, \mathbf{q}, A) p(\mathbf{w} \mid \mathbf{W}, y, m^*, \mathbf{q}, A) \, \mathrm{d}\mathbf{w} \\
&= \int \frac{p(y, m^* \mid x, \mathbf{w}, \mathbf{q}, A) p(x \mid \mathbf{w}, \mathbf{q}, A)}{p(y, m^* \mid \mathbf{w}, \mathbf{W}, \mathbf{q}, A)} p(\mathbf{w} \mid \mathbf{W}, y, m^*, \mathbf{q}, A) \, \mathrm{d}\mathbf{w},
\end{aligned}$$

Then, substituting in

$$\begin{aligned}
p(\mathbf{w} \mid \mathbf{W}, y, m^*, \mathbf{q}, A) &= \frac{p(\mathbf{w}, y, m^* \mid \mathbf{W}, \mathbf{q}, A)}{p(y, m^* \mid \mathbf{W}, \mathbf{q}, A)} \\
&= \frac{p(y, m^* \mid \mathbf{w}, \mathbf{W}, \mathbf{q}, A) p(\mathbf{w} \mid \mathbf{W})}{p(y, m^* \mid \mathbf{W}, \mathbf{q}, A)},
\end{aligned}$$

and cancelling gives

$$\begin{aligned}
\tilde{q}_x &= \int \frac{p(y, m^* \mid x, \mathbf{w}, \mathbf{q}, A) p(x \mid \mathbf{w}, \mathbf{q}, A) p(\mathbf{w} \mid \mathbf{W})}{p(y, m^* \mid \mathbf{W}, \mathbf{q}, A)} \, \mathrm{d}\mathbf{w} \\
&= q_x \int \frac{p(y, m^* \mid x, \mathbf{w}, \mathbf{q}, A) p(\mathbf{w} \mid \mathbf{W})}{\sum_{\tilde{x}} q_{\tilde{x}} \int p(y, m^* \mid \tilde{x}, \tilde{\mathbf{w}}, \mathbf{q}, A) p(\tilde{\mathbf{w}} \mid \mathbf{W}) \, \mathrm{d}\tilde{\mathbf{w}}} \, \mathrm{d}\mathbf{w}, \qquad (5)
\end{aligned}$$

where the last step uses $p(x \mid \mathbf{w}, \mathbf{q}, A) = p(x \mid \mathbf{q}) = q_x$.

It remains to find $\int p(y, m^* \mid x, \mathbf{w}, \mathbf{q}, A) p(\mathbf{w} \mid \mathbf{W}) \, \mathrm{d}\mathbf{w}$. Under PCM this is easily found since, given x, the probability of clicking any arm a considered by the user is the same as its independent click probability (as though it were the only arm in the set) and is independent from all weights except $w_{a,x}$. That is, for a single arm a,

$$\begin{aligned}\int p(y, m^* \mid x, \mathbf{w}, \mathbf{q}, A) p(\mathbf{w} \mid \mathbf{W})\, \mathrm{d}\mathbf{w} &= \int p(y, m^* \mid x, w_{a,x}, \mathbf{q}, A) p(w_{a,x} \mid W_{a,x})\, \mathrm{d}w_{a,x} \\ &= (\mu_{a,x})^y (1-\mu_{a,x})^{(1-y)}, \qquad (6)\end{aligned}$$

where $\mu_{a,x} = \frac{\alpha_{a,x}}{\alpha_{a,x}+\beta_{a,x}}$ is the expectation of $W_{a,x}$. Under PCM, the outcome of any arm, given it is considered by the user, is independent of the other arms so (5) and (6) can be combined to give

$$\tilde{q}_x = q_x \frac{(\mu_{a_{m^*},x})^y \prod_{a \in A'} (1-\mu_{a,x})}{\sum_{j=1}^{n} [q_j (\mu_{a_{m^*},x})^y \prod_{a \in A'} (1-\mu_{a,x})]}\,.$$

A.3 Updating for TCM with Known x

Adapting (4) in Section A.1, the updating for known x under TCM is

$$\begin{aligned}p(\mathbf{w}_{\mathcal{A}} | y, m^*, x) &= \frac{p(\mathbf{w}_{\mathcal{A}})}{p(y, m^* | \mathbf{q})} \int_{u=0}^{1} q_x (\mathbb{1}_{\{w_{a_{m^*},x} > u\}})^y \prod_{a \in A'} \mathbb{1}_{\{w_{a,x} \le u\}} du \\ &= \frac{q_x p(\mathbf{w}_{\mathcal{A}})}{p(y, m^* | \mathbf{q})} \int_{u=0}^{1} (\mathbb{1}_{\{w_{a_{m^*},x} > u\}})^y \mathbb{1}_{\{u > \max_{a \in A'} (w_{a,x})\}} du \\ &= \frac{q_x p(\mathbf{w}_{\mathcal{A}})}{p(y, m^* | \mathbf{q})} \left[(w_{a_{m^*},x})^y - \max_{a \in A'} (w_{a,x}) \right].\end{aligned}$$

B Lemma B.1

The following lemma is used in the proof of Theorem 5.1 which is given in Appendix C. Both this lemma and the proof of Theorem 5.1 use the following notation.

Let $R^{TS}(a, \mathbf{W}_{a,t}, \mathbf{q}_t \mid I_t) = \mathbf{q}_t \cdot \tilde{\mathbf{w}}_{a,t}$ denote the stochastic index for the multiple action Thompson sampling policy for a single arm a where each $\tilde{w}_{a,t,x} \sim W_{a,t,x}$. Then under SEQ the arm chosen in slot one is the one with the highest index: $a_{t,1} = \mathrm{argmax}_{a \in \mathcal{A}} R^{TS}(a, \mathbf{W}_{a,t}, \mathbf{q}_t | I_t)$.

Lemma B.1. *Let $\tau_{a,T}$ be the set of times $t = 1, \ldots, T$ at which $a \in A_t$. Let $q^* = \inf_{t,x} \Pr(q_{t,x} = 1)$ and $w^* = \max_{a \in \mathcal{A}, x} w_{a,x}$ and, from these, set $\eta = q^*(1-w^*)^m$. If $q^* > 0$ then under the deterministic updating scheme given in Section 4.2 using any click model from Section 3.2,*

$$\Pr\left(R^{TS}(a, \mathbf{W}_{a,T}, \mathbf{q}_T \mid I_t) \le \frac{1}{1+\eta-\delta_1} + \delta_2 \right) \to 1 \quad \text{as } |\tau_{a,T}| \to \infty$$

for any $a \in \mathcal{A}$ and any δ_1, δ_2 such that $\eta > \delta_1 > 0$ and $\delta_2 > 0$.

Proof. For any $a \in \mathcal{A}$, $x = 1, \dots, n$ we will give bounds for expected rate at which $\alpha_{a,t,x}$ and $\beta_{a,t,x}$ increase as the arm a is selected over time (an upper bound for $\alpha_{a,t,x}$ and a lower bound for $\beta_{a,t,x}$). This will give an asymptotic upper bound less than 1 on each posterior mean $\mu_{a,t,x} = \mathbb{E}\left[W_{a,t,x}\right]$ as $|\tau_{a,t}| \to \infty$. Showing that $Var(W_{a,t,x}) \to 0$ as $|\tau_{a,t}| \to \infty$ then gives the required result. Throughout, a is an arbitrary arm in $\mathcal{A}$ and x an arbitrary state in $\{1, \dots, n\}$.

Let $\alpha_{a,0,x}$ and $\beta_{a,0,x}$ be values of the parameters of the Beta prior placed on $w_{a,x}$, then an upper bound for $\alpha_{a,T,x}$, $T \geq 1$ is simply

$$\alpha_{a,T,x} \leq \alpha_{a,0,x} + |\tau_{a,T}| \tag{7}$$

since $\alpha_{a,T,x}$ can only increase by at most one at times when when $a \in A_t$ and is unchanging at other times.

For a lower bound on $\mathbb{E}\left[\beta_{a,T,x}\right]$ we consider only times when $a \in A_t$, $q_{t,x} = 1$ and $y_t = 0$. Then $y_t = 0$ guarantees that arm a is considered by the user and $q_{t,x} = 1$ means the failure to click can be attributed to $w_{a,x}$. Hence, for $t \geq 1$,

$$\beta_{a,t+1,x} \mid (q_{t,x} = 1, y_t = 0, a \in A_t, \beta_{a,t,x}) = \beta_{a,t,x} + 1. \tag{8}$$

At all times $\beta_{a,t+1,x} \geq \beta_{a,t,x}$ since the β parameters cannot decrease. For PCM,

$$\Pr(y_t = 0 \mid q_{t,x} = 1, A_t, \mathbf{w}_{A_t}) = \prod_{b \in A_t} (1 - w_{b,x})$$

which is no larger than the corresponding probability for TCM. The probability that $y_t = 0$ can therefore be bounded below. Let $w^* = \max_{b \in \mathcal{A},x} w_{b,x}$ and $q^* = \min_{t,x} \Pr(q_{t,x} = 1)$ then for any $A_t \subset \mathcal{A}$,

$$\Pr(y_t = 0 \mid A_t, \mathbf{w}_{\mathcal{A}}) \geq q^*(1 - w^*)^m. \tag{9}$$

We can now give a lower bound on $\mathbb{E}\left[\beta_{a,T,x} \mid I_1\right]$ where the expectation is joint over all $\mathbf{q}_t$, y_t, m_t^* for $t = 1, \dots, T$, and I_1 is just the priors for $\mathbf{W}$. Using (8) and (9), we have at any time T,

$$\begin{aligned}
\mathbb{E}[\beta_{a,T,x} \mid I_1] &\geq \beta_{a,0,x} + \sum_{t \in \tau_{a,T}} \Big[\Pr(q_{t,x} = 1) \Pr(y_t = 0 \mid q_{t,x} = 1, a \in A_t, \mathbf{w}_{A_t}) \Big] \\
&\geq |\tau_{a,T}| q^*(1 - w^*)^m. \qquad (10)
\end{aligned}$$

Let $\eta = q^*(1 - w^*)^m$ and note that $\eta > 0$ since $w^* < 1$ by the problem definition and $q^* > 0$ by the assumption given in the statement of the Lemma. Combining (7) and (10) gives, for any $\tau_{a,T}$,

$$\begin{aligned}
\mathbb{E}\left[\frac{\beta_{a,T,x}}{\alpha_{a,T,x}} \mid I_1\right] &\geq \frac{1}{\alpha_{a,0,x} + |\tau_{a,T}|} \mathbb{E}\left[\beta_{a,T,x} \mid I_1\right] \\
&\geq \frac{|\tau_{a,T}| \eta}{\alpha_{a,0,x} + |\tau_{a,T}|}
\end{aligned}$$

and so by the strong law of large numbers, for sufficiently large $|\tau_{a,T}|$ and conditional on I_1,

$$\frac{\beta_{a,T,x}}{\alpha_{a,T,x}} \geq \frac{|\tau_{a,T}|\eta}{\alpha_{a,0,x} + |\tau_{a,T}|} \to \eta \ . \tag{11}$$

Note that

$$\mu_{a,T,x} = \frac{\alpha_{a,T,x}}{\alpha_{a,T,x} + \beta_{a,T,x}} = \frac{1}{1 + \frac{\beta_{a,T,x}}{\alpha_{a,T,x}}} \ ,$$

and so from (11),

$$\Pr\left(\mu_{a,T,x} \leq \frac{1}{1+\eta-\delta_1}\right) \to 1 \quad \text{as} \quad |\tau_{a,T}| \to \infty \tag{12}$$

for any δ_1 such that $\eta > \delta_1 > 0$.

Then, using the variance of a Beta distribution and (10) we have

$$\begin{aligned} Var(W_{a,T,x}) &= \frac{\alpha_{a,T,x}\beta_{a,T,x}}{(\alpha_{a,T,x}+\beta_{a,T,x})^2(\alpha_{a,T,x}+\beta_{a,T,x}+1)} \\ &< \frac{(\alpha_{a,T,x}+\beta_{a,T,x})^2}{(\alpha_{a,T,x}+\beta_{a,T,x})^2(\alpha_{a,T,x}+\beta_{a,T,x}+1)} \\ &= \frac{1}{(\alpha_{a,T,x}+\beta_{a,T,x}+1)} \to 0 \quad \text{as} \quad |\tau_{a,T}| \to \infty, \end{aligned}$$

and so for any $\delta_2 > 0$ the sampled $\tilde{w}_{a,T,x} \sim W_{a,T,x}$ satisfy

$$\Pr\left(\tilde{w}_{a,T,x} \leq \mu_{a,T,x} + \delta_2\right) \to 1 \quad \text{as} \quad |\tau_{a,T}| \to \infty. \tag{13}$$

By definition $R^{TS}(a, \mathbf{W}_{a,t}, \mathbf{q}_t \mid I_t) = \sum_{x=1}^{n}(q_{t,x}\tilde{w}_{a,t,x}) \leq \max_x \tilde{w}_{a,t,x}$ where $\tilde{w}_{a,t,x} \sim W_{a,t,x}$. Therefore, to complete the proof it is sufficient that $\Pr(\tilde{w}_{a,T,x} < 1/(1+\eta-\delta_1)+\delta_2) \to 1$ as $|\tau_{a,T}| \to \infty$ for all $a \in \mathcal{A}$, $x = 1, \ldots n$ and any δ_1, δ_2 such that $\eta > \delta_1 > 0$ and $\delta_2 > 0$, which follows from (12) and (13). □

C Proof of Theorem 5.1

The proof will assume that there is a non-empty set of arms $\mathcal{A}_F \subset \mathcal{A}$ whose members are sampled finitely often as $t \to \infty$ and show that this leads to a contradiction. Under this assumption $\sum_{b\in\mathcal{A}_F} |\tau_{b,\infty}| < \infty$ and so there exists a finite time $M = \max_{b\in\mathcal{A}_F} \tau_{b,t}$ even as $t \to \infty$.

Let $\mathcal{A}_I = \mathcal{A} \setminus \mathcal{A}_F$ be the set of arms sampled infinitely often (which must be non-empty). Let $w^* = \max_{a\in\mathcal{A},x} w_{a,x}$ and $\eta = q^*(1-w^*)^m$ as in the proof of Lemma B.1. Note that $\eta > 0$ since $w^* < 1$ by the problem definition and $q^* > 0$ by the given condition. Then fix some $0 < \delta_1 < \eta$ and $0 < \delta_2 < 1 - 1/(1+\eta-\delta_1)$. Then by Lemma B.1 for all $a \in \mathcal{A}_I$,

$$\Pr\left(R^{TS}(a, \mathbf{W}_{a,t}, \mathbf{q}_t) \leq \frac{1}{1+\eta-\delta_1} + \delta_2\right) \to 1 \text{ as } t \to \infty.$$

So there exists a finite random time $T > M$ such that

$$\Pr\left(R^{TS}(a, \mathbf{W}_{a,t}, \mathbf{q}_t) \leq \frac{1}{1+\eta-\delta_1} + \delta_2\right) > 1 - \delta_2 \text{ for } t > T,\ \forall a \in \mathcal{A}_I. \tag{14}$$

Let $\epsilon = \min_{b \in \mathcal{A}_F}[\Pr(R^{TS}(b, \mathbf{W}_{b,T}, \mathbf{q}_T \mid I_T) > 1/(1+\eta-\delta_1) + \delta_2)]$. Then for all $t > T$, $b \in \mathcal{A}_F$ we have

$$\Pr\left(R^{TS}(b, \mathbf{W}_{b,t}, \mathbf{q}_t \mid I_t) > \frac{1}{1+\eta-\delta_1} + \delta_2\right) \geq \epsilon, \tag{15}$$

since no arm in $\mathcal{A}_F$ is selected at times $t > T > M$ and so $\mathbf{W}_{b,t}$ is unchanged over these times. We know that $\epsilon > 0$ since $\Pr(\tilde{w}_{b,T,x} > 1/(1+\eta+\delta_1) + \delta_2) > 0$ for all b, x because $1/(1+\eta-\delta_1) + \delta_2 < 1$ and $W_{b,T,x}$ is a Beta distribution with support $(0, 1)$.

Combining (14) and (15),

$$\Pr\left[R^{TS}(b, \mathbf{W}_{b,t}, \mathbf{q}_t \mid I_t) > R^{TS}(a, \mathbf{W}_{a,t}, \mathbf{q}_t \mid I_t), \forall a \in A\right] > \epsilon(1 - \delta_2) \tag{16}$$

for all $t > T$. Therefore

$$\sum_{t=T}^{\infty} \Pr(b \in A_t \text{ for some } b \in \mathcal{A}_F) > \sum_{t=T}^{\infty} \epsilon(1-\delta_2)^{|\mathcal{A}_I|} = \infty.$$

Using the Extended Borel-Cantelli Lemma (Corollary 5.29 of Breiman 1992) it follows that $\sum_{b \in \mathcal{A}_F} |\tau_{b,\infty}| = \infty$ which contradicts the assumption that $|\tau_{b,\infty}|$ is finite for all $b \in \mathcal{A}_F$. Therefore some arm in $\mathcal{A}_F$ is selected infinitely often and since $\mathcal{A}_F$ was of arbitrary size it follows that $\mathcal{A}_F = \emptyset$.

References

Agrawal, R., Gollapudi, S., Halverson, A. & Ieong, S. (2009), Diversifying search results, *in* ‘ACM Conference on Web Search and Data Mining (WSDM)’, ACM, pp. 5–14.

Ahmed, M. T. & Kwon, C. (2014), ‘Optimal contract-sizing in online display advertising for publishers with regret considerations’, *Omega* **42**(1), 201–212.

Auer, P. (2002), ‘Using confidence bounds for exploitation-exploration trade-offs’, *The Journal of Machine Learning Research* **3**, 397–422.

Auer, P., Cesa-Bianchi, N. & Fischer, P. (2002), ‘Finite-time analysis of the multiarmed bandit problem’, *Machine Learning* **47**(2), 235–256.

Balseiro, S. R., Feldman, J., Mirrokni, V. & Muthukrishnan, S. (2014), ‘Yield optimization of display advertising with ad exchange’, *Management Science* **60**(12), 2886–2907.

Breiman, L. (1992), *Probability*, Society for Industrial and Applied Mathematics, Philadelphia, PA.

Cappé, O. & Moulines, E. (2009), 'On-line expectation–maximization algorithm for latent data models', *Journal of the Royal Statistical Society, B* **71**(3), 593–613.

Chapelle, O., Manavoglu, E. & Rosales, R. (2015), 'Simple and scalable response prediction for display advertising', *ACM Transactions on Intelligent Systems and Technology (TIST)* **5**(4), 61.

Chen, W., Wang, Y. & Yuan, Y. (2013), Combinatorial multi-armed bandit: General framework and applications, *in* 'Proceedings of the 30th International Conference on Machine Learning', pp. 151–159.

Chen, Y.-J. (2017), 'Optimal dynamic auctions for display advertising', *Operations Research* **65**(4), 897–913.

Chen, Y., Pavlov, D. & Canny, J. F. (2009), Large-scale behavioral targeting, *in* 'Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 209–218.

Chuklin, A., Markov, I. & Rijke, M. d. (2015), *Click models for web search*, Vol. 7, Morgan & Claypool Publishers.

Craswell, N., Zoeter, O., Taylor, M. & Ramsey, B. (2008), An experimental comparison of click position-bias models, *in* 'Proceedings of the International Conference on Web Search and Web Data Mining', ACM, pp. 87–94.

Edwards, J. A. & Leslie, D. S. (2018), *Diversity as a Response to User Preference Uncertainty*, World Scientific, chapter 4, pp. 55–68.

El-Arini, K., Veda, G., Shahaf, D. & Guestrin, C. (2009), Turning down the noise in the blogosphere, *in* 'ACM Conference on Knowledge Discovery and Data Mining', ACM, pp. 289–298.

Fridgeirsdottir, K. & Najafi-Asadolahi, S. (2018), 'Cost-per-impression pricing for display advertising', *Operations Research* **66**(3), 653–672.

Gittins, J. C., Glazebrook, K. D. & Weber, R. (2011), *Multi-armed bandit allocation indices*, second edn, John Wiley & Sons, Chichester, UK.

Graepel, T., Candela, J. Q., Borchert, T. & Herbrich, R. (2010), Web-scale Bayesian click-through rate prediction for sponsored search advertising in Microsoft's Bing search engine, *in* 'Proceedings of the 27th International Conference on Machine Learning', pp. 13–20.

Hauser, J. R., Urban, G. L., Liberali, G. & Braun, M. (2009), 'Website morphing', *Marketing Science* **28**(2), 202–223.

Hillard, D., Schroedl, S., Manavoglu, E., Raghavan, H. & Leggetter, C. (2010), Improving ad relevance in sponsored search, *in* 'Proceedings of the third ACM international conference on Web search and data mining', ACM, pp. 361–370.

Hojjat, A., Turner, J., Cetintas, S. & Yang, J. (2017), 'A unified framework for the scheduling of guaranteed targeted display advertising under reach and frequency requirements', *Operations Research* **65**(2), 289–313.

Larsen, T., Leslie, D. S., Collins, E. J. & Bogacz, R. (2010), 'Posterior weighted reinforcement learning with state uncertainty', *Neural Computation* **22**(5), 1149–1179.

Li, L., Chu, W., Langford, J. & Schapire, R. E. (2010), A contextual-bandit approach to personalized news article recommendation, *in* 'Proceedings of the 19th International Conference on World Wide Web', ACM, pp. 661–670.

May, B. C., Korda, N., Lee, A. & Leslie, D. S. (2012), 'Optimistic Bayesian sampling in contextual-bandit problems', *Journal of Machine Learning Research* **13**(1), 2069–2106.

McMahan, H. B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Golovin, D., Chikkerur, S., Liu, D., Wattenberg, M., Hrafnkelsson, A. M., Boulos, T. & Kubica, J. (2013), Ad click prediction: a view from the trenches, *in* 'Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 1222–1230.

Najafi-Asadolahi, S. & Fridgeirsdottir, K. (2014), 'Cost-per-click pricing for display advertising', *Manufacturing & Service Operations Management* **16**(4), 482–497.

Nemhauser, G. L. & Wolsey, L. A. (1978), 'Best algorithms for approximating the maximum of a submodular set function', *Mathematics of Operations Research* **3**(3), 177–188.

Radlinski, F., Kleinberg, R. & Joachims, T. (2008), Learning diverse rankings with multi-armed bandits, *in* 'Proceedings of the 25th international conference on Machine learning', ACM, pp. 784–791.

Ricci, F., Rokach, L. & Shapira, B. (2011), *Recommender Systems Handbook*, Springer US, Boston, MA.

Richardson, M., Dominowska, E. & Ragno, R. (2007), Predicting clicks: estimating the click-through rate for new ads, *in* 'Proceedings of the 16th international conference on World Wide Web', ACM, pp. 521–530.

Rusmevichientong, P. & Williamson, D. P. (2006), An adaptive algorithm for selecting profitable keywords for search-based advertising services, *in* 'Proceedings of the 7th ACM Conference on Electronic Commerce', ACM, pp. 260–269.

Russo, D. & Van Roy, B. (2014), 'Learning to optimize via posterior sampling', *Mathematics of Operations Research* **39**(4), 1221–1243.

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.

Streeter, M., Golovin, D. & Krause, A. (2009), Online learning of assignments, *in* 'Advances in Neural Information Processing Systems', pp. 1794–1802.

Whittle, P. (1988), 'Restless bandits: Activity allocation in a changing world', *Journal of Applied Probability* **25**, 287–298.

Yan, J., Liu, N., Wang, G., Zhang, W., Jiang, Y. & Chen, Z. (2009), How much can behavioral targeting help online advertising?, *in* 'Proceedings of the 18th international conference on World wide web', ACM, pp. 261–270.

Yue, Y. & Guestrin, C. (2011), Linear submodular bandits and their application to diversified retrieval, *in* 'Advances in Neural Information Processing Systems', pp. 2483–2491.