Article:
Losada, D.E., Elsweiler, D., Harvey, M. et al. (2022) A day at the races : using best arm identification algorithms to reduce the cost of information retrieval user studies. Applied Intelligence, 52 (5). pp. 5617-5632. ISSN: 0924-669X

https://doi.org/10.1007/s10489-021-02719-2

Applied Intelligence
https://doi.org/10.1007/s10489-021-02719-2

A day at the races

Using best arm identification algorithms to reduce the cost of information retrieval user studies

David E. Losada[1] · David Elsweiler[2] · Morgan Harvey[3] · Christoph Trattner[4]

Accepted: 26 July 2021

Abstract

Two major barriers to conducting user studies are the costs involved in recruiting participants and researcher time in performing studies. Typical solutions are to study convenience samples or design studies that can be deployed on crowdsourcing platforms. Both solutions have benefits but also drawbacks. Even in cases where these approaches make sense, it is still reasonable to ask whether we are using our resources – participants' and our time – efficiently and whether we can do better. Typically user studies compare randomly-assigned experimental conditions, such that a uniform number of opportunities are assigned to each condition. This sampling approach, as has been demonstrated in clinical trials, is sub-optimal. The goal of many Information Retrieval (IR) user studies is to determine which strategy (e.g., behaviour or system) performs the best. In such a setup, it is not wise to waste participant and researcher time and money on conditions that are obviously inferior. In this work we explore whether Best Arm Identification (BAI) algorithms provide a natural solution to this problem. BAI methods are a class of Multi-armed Bandits (MABs) where the only goal is to output a recommended arm and the algorithms are evaluated by the average payoff of the recommended arm. Using three datasets associated with previously published IR-related user studies and a series of simulations, we test the extent to which the cost required to run user studies can be reduced by employing BAI methods. Our results suggest that some BAI instances (*racing algorithms*) are promising devices to reduce the cost of user studies. One of the racing algorithms studied, Hoeffding, holds particular promise. This algorithm offered consistent savings across both the real and simulated data sets and only extremely rarely returned a result inconsistent with the result of the full trial. We believe the results can have an important impact on the way research is performed in this field. The results show that the conditions assigned to participants could be dynamically changed, automatically, to make efficient use of participant and experimenter time.

Keywords Best arm identification · User studies · Racing algorithms

1 Introduction

Experimentation has become the most common research method in Library and Information Science (LIS) [15] and, in IR in particular, has a dominant empirical tradition [30]. It is only really in the last 10-15 years, however, that user studies such as controlled laboratory studies with human users have been commonly accepted as part of the programme at the premier IR conference, ACM SIGIR. The acceptance of this kind of empirical contribution resulted from a growing movement within LIS, e.g., [17, 34], but also from the increased recognition that such studies provide value complementary to traditional Cranfield experiments [40]. Laboratory-based user studies offer the possibility to learn about aspects, such as interaction, which are difficult to study using Cranfield experiments alone [61]. They also provide insight on how behaviours differ across groups (e.g., experienced vs inexperienced users [4]) and contexts (e.g., varying topic familiarity [41]).

✉ David E. Losada
david.losada@usc.es

[1] Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Santiago de Compostela, Spain

[2] University of Regensburg, Regensburg, Germany

[3] University of Sheffield, Sheffield, UK

[4] University of Bergen, Bergen, Norway

Published online: 17 August 2021

Despite now being commonly accepted in IR, user studies are often –sometimes unfairly– criticised for sample size, regardless of whether they are representative of the population studied or provide sufficient statistical power [12]. There are many reasons for small sample sizes in such studies. Recruitment is challenging, particularly when the population of interest includes highly-paid individuals with little time (e.g., lawyers [44], engineers [20] or healthcare professionals [27]). Moreover, each participant takes considerable time and effort to process; with informed consent and debrief this can take up to several hours each. A further issue is that the cost associated with running multiple conditions typically means reducing the number of conditions for reasons of pragmatism. There is a need to reduce the cost of user studies, not only in IR but also in other fields, where multiple user-related aspects are often studied [51, 65]. Even if power analysis is effectively employed to estimate the necessary number of participants and individual experiments required prior to the studies taking place, there can still be many instances where more resources are actually used than is necessary.

In this work we explore one means of using acquired participants, their time, and ours more efficiently. If we can achieve this, it may lower the entry barrier to user studies being performed in our field or perhaps allow additional conditions to be tested using the same resources. The idea is: rather than distributing the conditions uniformly across participants or participant tasks, as is typically done, we formulate the distribution of experimental conditions as an explore-exploit trade-off. We posit that, during the course of a study, experimenters incrementally gain information about which conditions are performing well and that this information could and should be used to design *adaptive user studies*. We treat the selection of conditions in a user study as a Best Arm Identification (BAI) problem and explore methods to intelligently adapt the adjudication of examples while a user study is in progress. If successful, this approach would offer a number of advantages, including: reduction in costs (due to the ability to run the study with fewer resources because less time is spent on poorly-performing conditions); effectiveness ([2] showed that uniformly allocating examples is a weak approach to correctly identifying the best performing model among a set of candidates); and user experience (the information obtained during the study is used to eliminate poorly-performing conditions and, thus, participants may potentially have a better user experience because they are not presented with inferior conditions).

While there is a history of adaptive trials in the testing of medical treatments (see review below), user studies in Information Retrieval (IR) and related disciplines do not presently utilise such approaches.

We explore here the value of a family of Multi-armed Bandit (MAB) algorithms –Best Arm Identification algorithms– to increase the efficiency of IR user studies. More specifically, our paper serves as a review of how existing BAI methods can alleviate the cost of certain types of user studies. BAI methods attempt to identify the best arm at a given confidence level, while consuming the minimum number of rounds. The motivation is that this framework provides a formal way to approach the problem of identifying the best experimental condition in the context of a user study whilst minimising the number of participants/individual experiments required. Each experimental condition is modelled as an arm in the BAI framework and the BAI algorithms provide us with a formal and effective way to guide the selection of the best condition. Using three freely-available data sets associated with previously published IR user studies, as well as a series of simulations, we test the extent to which the costs incurred (i.e. number of data points required to be collected) can be reduced.

2 Literature review

We review three bodies of related work. First, we report on methods for sampling users and determining appropriate sample sizes. Next, we review the use of adaptive trials, which have been used in medicine and other fields and for which we foresee benefit in IR. Lastly, we summarise MAB usage in A/B testing in our field, which is somewhat similar in concept and from which other types of user studies can draw inspiration.

2.1 Sampling approaches

One critical decision researchers must make when designing laboratory experiments with users is deciding how many participants to study. Most researchers who perform user studies are familiar with reviewer comments criticising sample size; reviewers use, sometimes incorrectly, sample size as a means to reject papers [12]. This phenomenon is known as the "sample size fallacy" and, although not often reported in the IR community, it has been described and empirically studied in other fields including HCI and medicine [6, 12, 29].

Acceptable sample size varies from field to field. In HCI, Nielsen controversially claimed that only 5 participants are needed for a qualitative usability study [56] and that 20 were sufficient for more quantitative studies as this "typically

offers a reasonably tight confidence interval" [55]. The first claim in particular has been disputed by others in the same field [63, 71]. A recent systematic review of user studies at the CHI 2014 conference found sample sizes ranging from 1 to 916,000 with a mean sample size for in-person laboratory studies of 20 (SD=12) [12]. In his tutorial for RecSys user studies Knijnenburg [43] is sceptical about the utility of small samples, which tend to be underpowered and are thus highly likely to miss important differences that exist. He is also critical of studies being overpowered, i.e. those that use a lot more resources than necessary. In interactive IR, the determination of sample size is often based on heuristics and limited by practical constraints such as time, availability of participants and finances [41]. As a result, many studies are underpowered. Sakai [62] performed post-hoc power analyses on 840 SIGIR full papers and 215 TOIS papers published between 2006 and 2015[1]. The analyses revealed that both highly overpowered and highly underpowered experiments are reported in the IR literature. While power analysis is recognised as a rigorous and defensible method of determining sample size, it is not without issue. One limitation is that it requires a pre-study understanding of effect size, which is often difficult or impossible to accurately estimate [25].

A second critical decision researchers must take relates to how participants are sourced. The difficulties in achieving appropriate sample sizes lead to sampling from participant pools that are not always representative of the target population. The use of convenience samples and over-representation of undergraduate students have raised some concerns about the external validity of experimental results in many fields [12, 22, 38, 47]. For HCI, Caine reports that 19% of studies examined reported college students as the sole participants. In political science, a review for the period 1990-2006 found that about a quarter of the reported experiments were based solely on student samples [57]. A further means by which sample sizes and sampling frames for convenient samples can be increased is to design a user study suitable for remote deployment [31, 42]. Despite the loss of environmental control and lack of ability to observe, the evidence suggests that many behaviours do not change significantly when studies are performed remotely rather than in the lab [42]. The approach can be taken further by using crowd-sourcing platforms, such as Amazon Mechanical Turk [45, 72], which have become increasingly popular in IS and IR studies.

Regardless of where participants are sourced they are a precious resource and their participation should not be taken for granted. We posit that adaptive trials may be a means of maximising the benefit of participant effort irrespective of study type. The following subsection reviews how adaptive trials have been used in the past.

[1]The papers included both batch analyses and user studies.

2.2 Adaptive trials

Although not applied in IR or related fields, the concept of adaptive trials has a long tradition in medicine [7, 16, 70, 75], where recruitment of study participants is even more challenging as they typically need to meet medical and geographical constraints. Moreover, randomly assigning patients to experimental conditions in clinical trials may have serious consequences. If researchers learn early in an experiment that a particular cancer treatment is more effective than a standard treatment they may feel ethically obliged to switch control group participants to the experimental condition as it may have existential outcomes. Several approaches have been proposed including techniques alluding to similar considerations as the multi-armed bandit problem. This includes play-the-winner strategies [75], drop-the-losers designs, where certain treatments are dropped or added in response to their response data [8], and Bayesian approaches, which choose the condition based on the highest posterior probability and can include stopping rules to facilitate early termination of a trial or condition, if appropriate [16, 36]. The guiding idea behind these is the ethical one of not prolonging a trial longer than necessary, as an unduly prolonged trial may result in an excessive number of patients being given the less beneficial treatment. See [14, 64] for detailed, recent reviews.

Aziz and colleagues [5] have worked on MAB designs for dose-finding in clinical trials. Their goal was to find the optimal dosage in early stage clinical trials. They tested multiple variants of Thompson Sampling and found solutions that outperform state-of-the-art dose identification algorithms. In the context of drug discovery, Terayama and colleagues [66] showed that a BAI algorithm was useful for structure-based drug design. The BAI method proposed by these authors can optimally control the number of simulations required to predict binding structures of drug candidate molecules. This team of researchers has also worked on how to effectively employ the BAI framework to select protein-protein complex structures [67].

In IR, assigning study participants to weaker conditions obviously has less grave consequences, the motivation for not wanting to prolong the experiment relates to cost and efficiency (we wish to achieve studies with

fewer resources or study more conditions with the same resources). Although untried in laboratory user studies, certain MAB techniques have been applied in our field for online controlled experiments, which use live systems. We summarise such work in the next sub-section.

2.3 MAB in IR evaluation

A wide range of bandit-based models have been employed to support tasks in multiple domains and applications [28, 46, 60]. MABs have been successfully employed in online IR experimentation. Online controlled experiments are now common when evaluating system effectiveness, particularly in industrial research contexts (e.g. [9, 58]). MAB algorithms have been used in this context to learn ranking strategies by minimising the total number of poor rankings displayed over time. This is a task which can be modelled conveniently as a explore-exploit trade-off [33, 53, 59]. The formulation of the problem and the type of MAB algorithms used vary. For example, Yue and Joachims [74] employed duelling bandits to learn from noisy, relative signals between two candidate rankers. Burtini et al. [11] surveyed MAB approaches useful for online experiment design. Note, however, that the approaches described above in relation to online evaluation differ from those in our context as these are typically k-armed problems aiming to minimise *total* regret. Such notion of regret is important in online studies because users use the system and should not be penalised or potentially lost (e.g., by offering poorer conditions). IR lab studies are typically different because participants are testing prototypes with simulated tasks provided by the experimenter and thus are neither penalised for poor outcomes nor are they really invested in the system's performance.

In the context of building test collections for batch evaluation of adhoc search, Losada and colleagues [48, 49] evaluated multiple bandit-based methods and concluded that a Bayesian approach performs the best at adjudicating judgments in pooling-based evaluation. Given a query and multiple search systems that contributed to the pool, a bandit-based solution iteratively learned about the quality of the systems and dynamically adapted the judgment process (by selectively choosing the systems from which new relevance judgments are made). This low-cost solution for creating relevance assessments was adopted by the TREC 2017 common core track [35]. Although these bandit-based models represent an example of effective use of MABs for reducing the cost in IR, they are intrinsically different to the BAI methods explored here. Losada and colleagues were interested in maximising the cumulative sum of rewards (i.e. number of relevant documents identified within the evaluation process) and, thus, they worked with several k-armed algorithms oriented to this task. BAI algorithms, instead, are oriented to minimise a notion of regret (see Section 3) that only depends on the quality of the final arm[2] (regardless of the rewards obtained within the process).

2.4 Contributions

The literature reviewed above has highlighted difficulties relating to recruitment and sample size in user studies and hinted that MABs may offer utility in such situations. Three forms of controlled study were mentioned (in-personal studies, remotely-deployed studies & crowd-sourced studies), all of which differ from the online evaluations summarised – for which MABs have been used in IR – in that a live system is not used. Below we study the benefits MABs might offer for these kinds of study using publicly-available user study data sets and a series of simulations.

More concretely we make the following contributions:

- We present the first investigation of the potential for BAI algorithms to reduce the cost of IR user studies
- We study the utility of common approaches on diverse data sets from the IR literature (spanning topics such as privacy, food search and recommendation), as well as synthetic data sets.
- We demonstrate that significant savings can be made (up to 72.4% fewer data points were achieved without any cost).
- We show that one algorithm Hoeffding offered consistent savings over both the real and simulated data sets.
- We present findings on how the scale of the study influences the benefit of the approaches demonstrating that advantages can be attained beyond 90 data points.

3 User studies as best arm identification problems

Let us consider the situation where researchers wish to evaluate different experimental conditions and need to identify the best performing one with respect to a single criterion. The researcher designs a user study (this could be in-person, remotely deployed or crowd-sourced), in which the conditions are tested by participants following either a between or within-groups design. Each participant performs one condition at a time and, in doing so, either implicitly or explicitly, provides a score (or his performance is evaluated using a given measure of performance). The goal of the user

[2]Formally speaking, the BAI algorithms evaluated here are *pure exploration algorithms*, while the methods tested by Losada et al. are classic MAB algorithms (see [2]).

study is to establish the experimental condition which is most likely to offer the highest overall performance.

Many IR user studies fit with the above description[3]. We posit that the problem of identifying the best performing experimental condition among a set of competing conditions can be naturally cast as a Best-Arm Identification problem in Multi-Armed Bandits. This is a forecasting task that can be solved in the context of MABs with independent arms (pulling one arm does not reveal any information about the other arms). Under this setting, multiple algorithms that implement some form of *gap-based* exploration have been developed [2]. Essentially, these consist of exploring the arms (i.e., the conditions) in order to reduce the uncertainty about the gaps between the rewards of the arms and, when there is sufficient confidence, output a recommended arm. Unlike standard MAB methods, where the goal is to maximise the cumulative rewards obtained, BAI methods are evaluated on the quality of the recommended arm at the end of exploration.

The general structure of a BAI problem is sketched in Algorithm 1, often referred to as the pure exploration problem [2]. The prediction is evaluated in terms of *regret*, which is the difference between the mean reward of the recommended arms and the mean reward of the optimal arm. BAI algorithms are also evaluated in terms of *sample complexity*, which is defined as the total number of rounds the algorithm performed before termination, and is clearly something we wish to minimise. Further details about complexity and BAI algorithms can be consulted in the work by Kaufmann and colleagues [26, 39], who have extensively worked on the characterization of the complexity of BAI algorithms.

Algorithm 1 Best arm identification.

input : K arms whose reward distributions $(\nu_1, \ldots, \nu_K)$ are unknown
output: Recommended arms $R \subseteq \{1, \ldots, K\}$
Loop
  Choose one or several arms
  Draw a reward from each selected arm
  Update the state (affects the next pick)
  if *STOPPING_CONDITION true* **then**
    Output a recommendation $R \subseteq \{1, \ldots, K\}$

In [2], the authors experimented with a number of simulated tasks and demonstrated that uniform arm allocation is substantially inferior to other alternatives. The results showed that the probability of error, defined as the probability of missing the optimal arm, is much smaller when the algorithm incorporates some form of bias towards the most effective arms. These experiments were performed under a wide range of conditions (different number of arms, varying difficulty –differences among the arms evaluated– and different number of rounds). These results inspired us to explore the role that BAI algorithms can play in optimising user studies. An intelligent selection of participant conditions may be beneficial, both in terms of cost (fewer number of rounds required to determine the optimal arm) and effectiveness (given the same budget, non-uniform alternatives have shown to be more precise). In the following, we explain the main characteristics of several algorithms that can be employed to support this task.

3.1 Racing algorithms

Racing algorithms, initially proposed by Maron and Moore [50], attempt to identify the best arm at a given confidence level while consuming the minimal number of rounds. To meet this aim, they quickly discard poor arms and concentrate effort on differentiating between the most promising ones. In practice, the algorithm is derived from Hoeffding's inequality [32], which defines the confidence in the sample mean of a series of independently drawn points.

We model the conditions as arms and employ BAI to quickly concentrate on the best conditions. Given K, the number of conditions, and N, the maximal number of rounds allowed for deciding, a racing algorithm either finishes when the rounds are exhausted or when it can state that, with probability at least $1 - \delta$, it has found the best condition.[4] Precisely, after any given number of plays ($t < N$) of a condition a, the following confidence interval is constructed for its mean reward:

$$\left[\mu_a - R\sqrt{\frac{log(2 \cdot K \cdot N/\delta)}{2t}}, \mu_a + R\sqrt{\frac{log(2 \cdot K \cdot N/\delta)}{2t}}\right] \tag{1}$$

where μ_a is the mean reward obtained from the t plays and R the range of the rewards obtained. In this way, each condition is associated with its estimated mean and Hoeffding's formula sets a bound on its possible spread. The main idea of the racing algorithm is to continuously

[3] We take three studies from the literature as a case in point.

[4] δ is the confidence level parameter, which we set to 0.05 in all the experiments.

eliminate those conditions whose best possible reward (upper bound) is still smaller than the worst possible reward of the best condition (lower bound). As more rounds are run, the intervals become smaller and the algorithm proceeds until it is left with a single condition or runs out of plays. The algorithm returns the condition(s) whose reward rates are insignificantly different after the whole process. Algorithm 2 sketches our implementation of the Racing Algorithm.

Algorithm 2 Racing algorithm (Hoeffding's race).

input : K conditions whose reward distributions $(v_1, \ldots, v_K)$ are unknown, δ (significance level parameter), N (maximum number of rounds)
output: Recommended conditions $R \subseteq \{1, \ldots, K\}$
$ActiveConditions \leftarrow \{1, \ldots, K\}$
$\mu \leftarrow \{0, \ldots, 0\}$
$Plays \leftarrow \{0, \ldots, 0\}$
$t \leftarrow 0$
Loop
  $t \leftarrow t + 1$
  for $a \in ActiveConditions$ **do**
    Draw a reward X_a from v_a
    Update μ_a and $Plays_a$
  $BestCondition \leftarrow \arg\max_k \mu_k$
  $LowBoundBestCondition \leftarrow \mu_{BestCondition} - R\sqrt{\frac{log(2 \cdot K \cdot N/\delta)}{2Plays_{BestCondition}}}$
  Remove from $ActiveConditions$ all conditions k where $\mu_k + R\sqrt{\frac{log(2 \cdot K \cdot N/\delta)}{2Plays_k}} < LowBoundBestCondition$
  if $|ActiveConditions| = 1$ OR $t = N$ **then**
    return($ActiveConditions$)

Alternative bounds to those set by Hoeffding's inequality were proposed in [3]. The so-called empirical Bernstein bounds incorporate variance information in a principled manner and quickly become much tighter than Hoeffding's bounds. The resulting confidence interval is:

$$\left[\mu_a - (\overline{\sigma_a}\sqrt{\frac{2 \cdot log(3 \cdot K \cdot N/\delta)}{t}} + \frac{3 \cdot R \cdot log(3 \cdot K \cdot N/\delta)}{t}), \right.$$
$$\left. \mu_a + (\overline{\sigma_a}\sqrt{\frac{2 \cdot log(3 \cdot K \cdot N/\delta)}{t}} + \frac{3 \cdot R \cdot log(3 \cdot K \cdot N/\delta)}{t})\right] \quad (2)$$

where $\overline{\sigma_a}$ is the empirical standard deviation of the observed rewards. This bound leads to an alternative Racing Algorithm [52], which is a variant of Algorithm 2 (where the Hoeffding bounds –(1)– are replaced by Bernstein bounds –(2)–). This variant will be referred to as Bernstein's Race.

3.2 Elimination algorithms

Even-Dar and colleagues [23, 24] proposed several Sucessive Elimination algorithms for the BAI problem, which repeatedly sample arms and eliminate the arm which has the lowest empirical reward in a principled manner. The resulting algorithm, illustrated in Algorithm 3, is guaranteed to select the optimal condition with probability at least δ. The number of steps taken (*sample complexity*) is bounded (see [24], Theorem 3).

Algorithm 3 Succesive elimination (SE) algorithm.

input : K conditions whose reward distributions $(v_1, \ldots, v_K)$ are unknown, δ (significance level parameter), N (maximum number of rounds)
output: Recommended conditions $R \subseteq \{1, \ldots, K\}$
$ActiveConditions \leftarrow \{1, \ldots, K\}$
$\mu \leftarrow \{0, \ldots, 0\}$
$t \leftarrow 0$
Loop
  $t \leftarrow t + 1$
  for $a \in ActiveArms$ **do**
    Draw a reward X_a from v_a
    Update μ_a
  $BestCondition \leftarrow \arg\max_k \mu_k$
  $\alpha \leftarrow \sqrt{\frac{\log(c \cdot K \cdot t^2/\delta)}{t}}$
  Remove from $ActiveConditions$ all conditions k where $\mu_{BestCondition} - \mu_k \geq 2 \cdot \alpha$
  if $|ActiveConditions| = 1$ OR $t = N$ **then**
    return($ActiveConditions$)

A second algorithm, Median Elimination (ME), has a better dependence on the number of arms and improves the sample complexity bound by a logarithmic factor. To meet this aim, the algorithm discards the worst half of the arms on each round. Algorithm 4 depicts this method. This algorithm outputs an ϵ-optimal condition, which is defined

as one whose expected reward is at most ϵ from the optimal reward.[5]

Algorithm 4 Median elimination (ME) algorithm.

input : K conditions whose reward distributions $(\nu_1, \ldots, \nu_K)$ are unknown, δ (significance level parameter), ϵ, N (maximum number of rounds)
output: Recommended conditions $R \subseteq \{1, \ldots, K\}$
$ActiveConditions \leftarrow \{1, \ldots, K\}$
$\mu \leftarrow \{0, \ldots, 0\}$
$\epsilon \leftarrow \epsilon/4$
$\delta \leftarrow \delta/2$
$t \leftarrow 0$
Loop
  $t \leftarrow t + 1$
  for $a \in ActiveConditions$ **do**
    Sample condition a for $\left\lfloor \frac{1}{(\epsilon/2)^2} ln(3/\delta) \right\rfloor$ times
    Update μ_a
  $MedianReward \leftarrow median(\mu)$
  Remove from $ActiveConditions$ all conditions k where $\mu_k < MedianReward$
  if $|ActiveConditions| = 1$ OR $t = N$ **then**
    return($ActiveConditions$)
  $\epsilon \leftarrow \frac{3}{4}\epsilon$
  $\delta \leftarrow \delta/2$

3.3 LUCB algorithm

Kalyanakrishnan and colleagues [37] designed an algorithm named LUCB that has improved sample complexity. The algorithm is inspired by the Upper Confidence Bound (UCB) algorithm, which has been popularly employed for regret minimisation in standard MAB problems. Elimination algorithms find it difficult to ensure low sample complexity because sometimes they induce erroneous eliminations. LUCB, instead, maintains a separation between the stopping rule and the sampling strategy and never eliminates any competing arm. Such an approach guarantees a low expected sample complexity.

[5]We experimented with the less stringent variant of Median Elimination (ϵ=1) because lower values of ϵ require a large sample from the conditions in the first execution of the for loop. Such large initial sample would prevent the use of this method for most user studies and, furthermore, we are interested here in the practical consequences of the use of the method rather than on its theoretical guarantees.

Algorithm 5 LUCB1 algorithm.

input : K conditions whose reward distributions $(\nu_1, \ldots, \nu_K)$ are unknown, δ (significance level parameter), ϵ, N (maximum number of rounds)
output: Recommended conditions $R \subseteq \{1, \ldots, K\}$
$\mu \leftarrow \{0, \ldots, 0\}$
$Plays \leftarrow \{0, \ldots, 0\}$
for $a \in \{1, \ldots, K\}$ **do**
  Draw a reward X_a from ν_a
  Update μ_a and $Plays_a$
$rounds \leftarrow 1$
Loop
  $BestCondition \leftarrow \arg\max_k \mu_k$
  $LowBoundBestCondition \leftarrow \mu_{BestCondition} - \sqrt{\frac{1}{2 \cdot Plays_{BestCondition}} \cdot ln(\frac{5 \cdot K \cdot rounds^4}{4 \cdot \delta})}$
  $HUCB\_Condition \leftarrow \arg\max_{k, k \neq BestCondition} \left(\mu_k + \sqrt{\frac{1}{2 \cdot Plays_k} \cdot ln(\frac{5 \cdot K \cdot rounds^4}{4 \cdot \delta})}\right)$
  $HighBoundHUCB\_Condition \leftarrow \left(\mu_{HUCB\_Condition} + \sqrt{\frac{1}{2 \cdot Plays_{HUCB\_Condition}} \cdot ln(\frac{5 \cdot K \cdot rounds^4}{4 \cdot \delta})}\right)$
  if $HighBoundHUCB\_Condition - LowBoundBestCondition < \epsilon$ OR $rounds = N$ **then**
    return($BestCondition$)
  Draw a reward $X_{BestCondition}$ from $\nu_{BestCondition}$
  Draw a reward $X_{HUCB\_Condition}$ from $\nu_{HUCB\_Condition}$
  Update $\mu_{BestCondition}$, $\mu_{HUCB\_Condition}$, $Plays_{BestCondition}$ and $Plays_{HUCB\_Condition}$
  $rounds \leftarrow rounds + 1$

The LUCB algorithm (Algorithm 5) proceeds as follows: First, the process is initialised by sampling each condition once. On each subsequent round, the algorithm extracts the best performing condition, estimates its lower performance bound and, subsequently, the competing condition with the highest upper confidence bound ($HUCB\_Condition$) is obtained. The algorithm stops when the difference between the highest upper bound of the competing conditions and the low bound of the best performing condition falls below ϵ[6]. If the algorithm does not stop then the method samples the conditions $BestCondition$ and $HUCB\_Condition$ and continues to the next round. The rationale is that it is

[6]In all our experiments we set ϵ to 0.1. This setting leads to a reasonable sample complexity (i.e., number of rounds, see [37], Theorem 6) and LUCB1 configured in this way can support typical user studies.

advisable to sample these two conditions instead of others as these represent the frontier between the best performing condition and the others.

4 Data

We conducted experiments with both data obtained from real user studies and data obtained from simulations. In doing so, we are able to evaluate the performance of the algorithms under real world conditions, as well as with varying levels of performance, which we can exactly control for the simulated data.

4.1 User studies

We chose to evaluate the algorithms on sets of real-world data from user studies described in recent publications related to Information Retrieval. The data sets were created by various authors on different IR-related topics, but all are freely available to download online. The studies differ considerably in their aims, conditions tested and methods to assess the quality of the conditions.

In line with our research aims, to be considered for our experiments the data set had to meet the following critiera: i) the data set must be sourced from an experiment involving human users relating to information retrieval, ii) multiple conditions are evaluated and compared (e.g., multiple search methods, interface designs, or summarization strategies) and iii) it must be possible to identify a clearly defined dependent variable associated with each condition (e.g., clicks or ratings from human users).

To perform BAI experiments on the selected data sets, we iteratively assigned rewards to the conditions based on the users' interactions.

- The first data set, from a recent ACM CHIIR paper by Zimmerman et al. [76], was collected by means of a controlled in-person laboratory study. The experiment studied user search behaviour for health-related information (n=40) and how this relates to privacy invasion. Four SERP variants were evaluated and the main aim was to determine the impact of these variants on good decision making and privacy protection. Performance was measured by the average number of privacy trackers encountered during searches. We modelled this user study as a 4-arm problem, where the arms (conditions) were *control*, *nudge_filter*, *nudge_rank* and *nudge_stoplight*. Every SERP, produced by a given condition, was assigned a non-binary reward (in [0,1]) based on the number of privacy trackers encountered (the fewer the better). We refer to this data set as **Privacy**.[7]
- The second data set, described in a recent ACM SIGIR paper by Elsweiler and colleagues [21], was collected by means of a remotely-deployed study performed as part of research work on helping people to make healthier food choices. Two algorithms were tested, *top10* and *images*, which used different features of online recipes to predict which of a given pair of recipes a user would most likely choose. These recipe pairs, of which there were 50, were chosen such that the two recipes were similar in terms of their constituent ingredients but had a large percentage difference in their fat content per 100g. Research shows that, given choices of otherwise similar food, people typically choose the fattier option. Participants (n=136) were shown pairs of recipes and asked to choose which one they would like to cook and eat. The model gets a reward if the user chooses the recipe in the pair with the least fat. We will model this user study as a 2-arm problem where rewards are binary and will refer to it as **Nudge**.[8]
- The other four data sets were collected using crowdsourcing by Trattner and Jannach as part of research work investigating the problem of similar item recommendation, a common feature of many websites which points users to other interesting objects given a currently inspected item [68]. This was investigated in two domains of "quality and taste" (recipes and movies). The main task given to participants was to individually assess five similar item recommendations for a given reference item. The study had two questions in the form of five-point Likert scales for each recommendation: i) the similarity between each recommendation and the reference item, and ii) how likely it is that they would try out each recommendation. The movies study had 12 recommendation strategies and the recipes study had 6 recommendation strategies. Given the data from these studies, we tested BAI algorithms i) to rapidly estimate the quality of the different strategies in terms of selecting similar items, and ii) to rapidly estimate the quality of the different strategies in terms of selecting items that the users are likely to try. Each data point is a recommendation list presented to the user and the associated reward is the aggregation of Likert responses on similarity or "likely to try", respectively (the five responses are added and the sum is divided by the maximum possible score). We refer to these data sets by combining each domain (**Movies** or **Recipes**) and each question (**Sim** or

[7] The data were provided by the authors via a GitHub repository (https://github.com/stevenzim/chiir-2019, last accessed December, 2019).

[8] The authors made the data available online (https://ai.ur.de/fibc/datasets.html), last accessed December, 2019).

Table 1 Statistics of user study data sets

User study	# Data points	Conditions (performance)
Privacy	320	control (0.87)
		nudge_filter (0.95)
		nudge_rank (0.96)
		nudge_stoplight (0.863)
Nudge	2,219	image (0.65)
		top-10 (0.59)
Movies-Sim	5,061	all (0.59)
		all-all (0.63)
		date (0.39)
		directors (0.47)
		genre (0.54)
		image (0.44)
		plot (0.50)
		rand (0.36)
		stars (0.45)
		svd (0.56)
		tag (0.62)
		title (0.41)
Movies-Try	5,061	all (0.61)
		all-all (0.62)
		date (0.53)
		directors (0.58)
		genre (0.58)
		image (0.53)
		plot (0.58)
		rand (0.52)
		stars (0.57)
		svd (0.63)
		tag (0.59)
		title (0.52)
Recipes-Sim	3,683	all (0.63)
		dir (0.58)
		img (0.50)
		ing (0.61)
		rand (0.38)
		title (0.58)
Recipes-Try	3,683	all (0.66)
		dir (0.64)
		img (0.62)
		ing (0.66)
		rand (0.58)
		title (0.65)

Try). For example, the similarity question – question i) – for the movies data is named **Movies-Sim**.[9]

Table 1 shows statistics for the six user study-derived data sets. Note that for each dataset a performance metric is calculated for all conditions. As the studies are different, the metric reported is different. In the case of **Nudge**, performance is based on a binary reward i.e. how often the condition led to the participant choosing the healthy choice of two recipes. In **Privacy**, the reward is a normalised value (ranging in [0, 1]) whereby a higher score reflects fewer trackers being accessed by participants. In all the experiments, the BAI algorithms were run on a random permutation of the available data points and each BAI algorithm was run until the best condition was chosen or until some condition exhausted its maximum number of points. For example, if we allow a maximum of 100 points per condition then we have to stop when any condition was tested 100 times (and recommend the condition with the highest performance so far). Observe that the BAI algorithms still make substantial savings in these cases because, unlike a full user study, they tend to quickly discard weak performing conditions, leading to savings in the overall effort.

This process was repeated 20 times (20 random sequences) and the results were averaged. The BAI algorithms are evaluated in terms of the percentage of savings (reduced effort with respect to the full user study) and the probability of error (normalised number of times where the BAI algorithm did not recommend the arm that had the highest performance in the full user study).

4.2 Simulated user study data

To further evaluate the BAI algorithms under different conditions, we performed additional experiments using sets of simulated data. Inspired by [2], we simulated K-arm problems where the conditions are modelled by probability distributions with rewards obtained by sampling from the distribution associated with each conditions. We generated 14 simulated datasets; 7 producing binary rewards (as in **Nudge**) and 7 producing non-binary rewards (as in the other real user studies). For all simulations, we generated K conditions and each condition was parameterized by $param_k$. The best condition had always the first index, and we set its parameter ($param_1$) to 0.5 (Bernoulli parameter

[9]The authors made the data available online (https://ai.ur.de/fibc/datasets.html), last accessed December, 2019).

or mean of the Truncated Normal set to 0.5, respectively). We then continued to generate performance for weaker conditions by varying the Bernoulli or Truncated Normal parameter as appropriate[10].

To test how different approaches function in diverse situations, we tailored the simulated experiments, such that each experiment corresponds to varying performance differences between conditions. As in [2], conditions were either clustered into groups or distributed according to an arithmetic or geometric progression. In doing so, we can represent divergent levels of difficulty for the BAI algorithms (i.e. the closer weaker conditions get to matching that of the best condition, the more difficult the task is for algorithms). The following experiments represent diverse plausible scenarios for IR user studies:

I. one group of weak conditions, $K = 20, \forall_{j=2..20}\ param_j = 0.3$.
II. two groups of weak conditions, $K = 20, \forall_{j=2..6}\ param_j = 0.33, \forall_{j=7..20}\ param_j = 0.27$.
III. geometric progression, $K = 4, \forall_{j=2..4}\ param_j = 0.5 - (0.47)^j$.
IV. 6 conditions divided in three groups, $K = 6, param_2 = 0.45, param_3 = param_4 = 0.35, param_5 = param_6 = 0.25$.
V. arithmetic progression, $K = 15, \forall_{j=2..15}\ param_j = 0.5 - (0.03) \cdot j$.
VI. two good conditions and a large group of weak conditions, $K = 20, param_2 = 0.48, \forall_{j=3..20}\ param_j = 0.27$.
VII. three groups of bad conditions, $K = 30, \forall_{j=2..6}\ param_j = 0.45, \forall_{j=7..20}\ param_j = 0.43, \forall_{j=21..30}\ param_j = 0.38$.

These seven experimental designs combined with the two alternative distributions (i.e. binary and non-binary) produce 14 different simulated scenarios. The number of samples produced from each condition was set to 1,000 for all simulated experiments. While 1,000 data points per condition is far from small, the real datasets described above show that this is not an implausible figure. Each BAI algorithm was run on a random permutation of the simulated data and the algorithm was run until either the best condition was found or some condition was exhausted. Each simulation was repeated 20 times and the results reported are averages of the 20 executions. The probability of error represents the proportion of cases were the BAI algorithm did not select the first condition.

[10] In all the experiments, the Truncated Normal Distributions had the standard deviation parameter set to 0.1.

5 Results

5.1 User study data

The first result to report is that on the **Privacy** data set, none of the algorithms offered any improvement. In each case, all 320 data points were required and as such, none of the algorithms stopped early. Large improvements were, however, found for the remaining real-world data sets. The results are summarised in Table 2, which reports the effort (#number of data points –pulls– required), the percentage of savings and the probability of generating an outcome different to that with the full data set. Hoeffding, Bernstein and Successive Elimination are all promising methods to reduce the cost of user studies without resulting in unacceptably high error rates. Bernstein and SE are the most conservative and, thus, save less –in some cases only reducing the number of necessary trials by a little less than 2%. However, these two methods have the same probability of error as Hoeffding, suggesting that it may be the most useful method overall. Observe that this method can produce up to a 15% reduction in cost whilst making nearly no mistakes; only for the **Movies-Try** data does the method sometimes identify a condition that is not the optimal. However, note (Table 1) that in **Movies-Try** the difference between the best condition (svd) and the second best (all-all) is negligible and, thus, arguably, selecting the second best is not a major issue. Indeed, even after running the full user study there is some uncertainty about the "true" winner. Observe also that, in practice, the recommendation of the BAI algorithm can be complemented with proper statistics for the competing conditions (e.g., confidence intervals after running the study) and, thus, the experimenter can gain further insights into the difference between the chosen condition and its competitors.

Despite LUCB1 being the algorithm that, overall, results in the least effort for all data sets except **Nudge**, offering up to $\sim$ 79% improvement, it also errs an unacceptably large number of times for the **Movies-Sim**, **Movies-Try** and **Recipes-Try** data sets. This means that, although it has the greatest potential for savings, it also has by far the greatest risk of incorrectly identifying the best condition. Median Elimination generally performs well by saving considerable effort (between 25 and 72% savings) whilst maintaining a low error rate. In the **Recipes-Try** and **Movie-Try** data sets, however, the error rate is unacceptably high at 0.4 and 0.2 respectively.

5.2 Simulated data

Results from the simulated data sets are described in Table 3. These generally align with those reported for the real-world data sets. Again, we find that LUCB1 offers

Table 2 Results - real user studies data

	Effort	% savings	Prob. error
Nudge			
Total Effort	2,219		
Hoeffding	2,176	1.94%	0.0
Bernstein	2,176	1.94%	0.0
SE	2,176	1.94%	0.0
ME	612	72.42%	0.0
LUCB1	2,219	0.0%	0.0
Movies-Sim			
Total Effort	5,061		
Hoeffding	4,598.75	9.13%	0.0
Bernstein	4,830.3	4.56%	0.0
SE	4,847	4.23%	0.0
ME	3,798	24.96%	0.0
LUCB1	1,078.9	78.68%	0.2
Movies-Try			
Total Effort	5,061		
Hoeffding	4,847	4.23%	0.05
Bernstein	4,847	4.23%	0.05
SE	4,847	4.23%	0.05
ME	3,798	24.96%	0.4
LUCB1	1,165.3	76.97%	0.3
Recipes-Sim			
Total Effort	3,683		
Hoeffding	3,129.55	15.03%	0.0
Bernstein	3,265.15	11.35%	0.0
SE	3,413	7.33%	0.0
ME	2,115	42.57%	0.05
LUCB1	1,398	62.04%	0.0
Recipes-Try			
Total Effort	3,683		
Hoeffding	3,413	7.33%	0.0
Bernstein	3,413	7.33%	0.0
SE	3,413	7.33%	0.0
ME	2,118.5	42.48%	0.2
LUCB1	1,691.2	54.08%	0.15

large savings, but is far too risky to be of use –in some cases erring more than half of the time and returning negligible (i.e. acceptable) error rates for only 3 out of the 14 simulations. In contrast to the real-world data, where it also tended to be somewhat error-prone, for the simulated data Median Elimination does not make any mistakes and is consistently able to reduce effort by around 35%.

Other findings of note include that Successive Elimination does not make mistakes but offers little benefit. Confirming the positive results from the real-world datasets, neither Bernstein nor Hoeffding provide different outcomes to the full data set in any of the experiments (i.e. do not err). Both, however, often offer substantial savings. Hoeffding tends to offer larger savings more often, particularly in the binary case. A general observation is that the methods tend to save more under non-binary situations. In all of the experiments, the variant that produced non-binary rewards led to higher rates of savings. This is likely because when rewards are non-binary there is greater scope to distinguish among the competing conditions.

To gain an understanding of what sizes of user study can benefit from BAI, we ran an experiment using Hoeffding with varying numbers of maximum numbers of samples to be produced from each condition (i.e. varying the number of points per condition). We experimented with 10 to 1000 (in steps of 10) points per condition, ran the simulation and recorded the point where the BAI started producing savings with respect to the full user study. Hoeffding was chosen as it offers consistently good performance in both the real and simulated experiments and makes almost no errors. We wanted to establish from when this algorithm starts to offer benefit. The results of these experiments are shown in Table 4. We tested the 14 simulated studies described above and, thus, we can see how different effect sizes (modelled by the 14 different configurations) behave with respect to the size of the user study (as the number of data points per condition directly determines the size of the full study). No figures are given for Experiment VII in Table 4 as Hoeffding offered no benefit at all in this scenario.

The results show that the starting point for benefits when applying Hoeffding vary with experiment, ranging from 90 (Experiment V, binary and non-binary) to 380 data points (Experiment I, non-binary) per condition. It seems that binary reward experiments saw benefit more quickly. Experiment V, where weak arms became progressively worse, saw the earliest benefit. Whereas, experiment I, which had a single group of weaker arms, saw the benefit come last. This makes sense because the worst performers of Experiment V have mean effectiveness scores (e.g., 0.05, 0.10, 0.15) that are substantially lower than the best arm's performance (0.5) and, thus, the BAI algorithm can quickly discard these low performers and, as a consequence, savings come earlier. The rest of the experiments exhibited profit after a comparable number of data points.

6 Discussion

We discuss our work in three sub-sections. First, we discuss what our findings mean with respect to the points outlined in the introduction and related work. Next, study limitations

Table 3 Results - simulated user studies

	Binary			Non-binary		
	Effort	% savings	PE	Effort	% savings	PE
Experiment I						
Total Effort	20,000					
Hoeffding	16.143,5	19.28%	0.0	11,180.65	44.10%	0.0
Bernstein	20,000	0.00%	0.0	11,326.45	43.37%	0.0
SE	20,000	0.00%	0.0	20,000	0.00%	0.0
ME	12,586	37.07%	0.0	12,430	37.85%	0.0
LUCB1	529.5	97.35%	0.0	2,018	89.91%	0.0
Experiment II						
Total Effort	20,000					
Hoeffding	19,953.15	0.23%	0.0	11,015.5	44.92%	0.0
Bernstein	20,000	0.00%	0.0	10,747.45	46.26%	0.0
SE	20,000	0.00%	0.0	20,000	0.00%	0.0
ME	12,523.6	37.38%	0.0	12,430	37.85%	0.0
LUCB1	2,018	89.91%	0.6	2,018	89.91%	0.05
Experiment III						
Total Effort	4,000					
Hoeffding	3,535	11.63%	0.0	3,361.4	15.97%	0.0
Bernstein	3,988.7	0.28%	0.0	3,351.55	16.21%	0.0
SE	4,000	0.00%	0.0	4,000	0.00%	0.0
ME	2,600.9	34.98%	0.0	2,542	36.45%	0.0
LUCB1	2,002	49.95%	0.05	2,002	49.95%	0.15
Experiment IV						
Total Effort	6,000					
Hoeffding	4,830.65	19.49%	0.0	4,247.6	29.21%	0.0
Bernstein	5,827.5	2.88%	0.0	4,077.15	32.05%	0.0
SE	6,000	0.00%	0.0	6,000	0.00%	0.0
ME	3,840.4	35.99%	0.0	3,778	37.03%	0.0
LUCB1	2,004	66.60%	0.6	2,004	66.60%	0.2
Experiment V						
Total Effort	15,000					
Hoeffding	8,764.65	41.57%	0.0	7,896.85	47.35%	0.0
Bernstein	12,373.5	17.51%	0.0	8,424.05	43.84%	0.0
SE	12,606.4	15.96%	0.0	13,039.1	13.07%	0.0
ME	9,652	35.65%	0.0	9,652	35.65%	0.0
LUCB1	2,013	86.58%	0.4	2,013	86.58%	0.15
Experiment VI						
Total Effort	20,000					
Hoeffding	13,520.1	32.40%	0.0	10,043.65	49.78%	0.0
Bernstein	19,999.5	0.00%	0.0	10,714.55	46.43%	0.0
SE	20,000	0.00%	0.0	20,000	0.00%	0.0
ME	12,679.6	36.60%	0.0	12,430	37.85%	0.0
LUCB1	2,018	89.91%	0.35	2,018	89.91%	0.20

Table 3 (continued)

	Binary			Non-binary		
	Effort	% savings	PE	Effort	% savings	PE
Experiment VII						
Total Effort	30,000					
Hoeffding	30,000	0.00%	0.0	30,000	0.00%	0.0
Bernstein	30,000	0.00%	0.0	29,964.45	0.12%	0.0
SE	30,000	0.00%	0.0	30,000	0.00%	0.0
ME	18,953.2	36.82%	0.0	18,610	37.97%	0.0
LUCB1	2,028	93.24%	0.4	2,028	93.24%	0.35

are discussed and, finally, we reflect on how the results may be utilised in practice.

6.1 Principal findings

The experimental results show that the kind of algorithms we have tested offer promise with respect to substantially lowering the entry barrier to performing user studies. We have shown empirically that data points can be saved; using the **Nudge** data set, median elimination used 72.4% fewer data points without incurring any error in the results. In several other cases, up to 38% savings were made whilst achieving the same results as if the full user study had been performed. These are considerable benefits which, depending on the study design, would translate into fewer participants being recruited, more conditions studied or individual participants being asked to do less work. Such differences could potentially mean less reliance on convenience samples, more user studies being performed or less fatigued participants. Moreover, unlike power analysis, no pre-study effect size estimate is needed.

The primary take-away from our results is that one of the racing algorithms, Hoeffding, holds the most promise. This algorithm offered consistent savings across both the real and simulated data sets. It only extremely rarely, as discussed above, returned a result inconsistent with the result of the full trial. Another important benefit of Hoeffding is that it only requires two input parameters (the significance level and the maximum number of rounds), while other algorithms, such as ME or LUCB1, also need ϵ, whose setting might not be obvious.

We emphasise that if a researcher wishes to perform a user study where the aim is to determine which empirical condition performs best (and has a single metric with which to measure performance), then there is no clear *disadvantage* to applying our adaptive approach, driven by the Hoeffding algorithm. The methods are simple to deploy and, even in cases where no gains were made – as in the **Privacy** data set – no costs are incurred. If, however, researchers wish to place a greater emphasis on recruitment savings (for example, when participants from the target population are extremely rare or expensive), then Median Elimination may be an option. This algorithm leads to savings that are typically larger than those achieved by Hoeffding. However, the researcher should be aware that in doing so the likelihood of attaining an incorrect result is increased.

The fact that no advantage was observed for the in-person lab study (**Privacy**) data set most likely results from the data set being too small to benefit. The earliest performance gain in the simulated experiments was observed from 90 data points per condition, which was beyond that of **Privacy** study. However, a between-groups design with 3 tasks per participant and $n > 30$ would in this case start leading to savings and anyone who has performed user studies will testify that, after having completed trials with 30 participants, savings are welcome. Given the number of data points required before benefits are seen, the results suggest that the approach is most useful for remotely-deployed and crowded-sourced studies. This setting would also be the easiest in which to build the algorithms into the process. It could be argued that lower costs associated with recruitment and performance in these types of study make the savings less pertinent. A counter argument would be that even in the case of crowd-sourcing, where costs are known to be particularly low, there are cases where potential participant pools are very small, such as studies of users with particular demographics, language skills or impairments [13].

6.2 Limitations

A number of limitations with the presented work are worthy of discussion. An obvious one is that despite the evidence provided for efficiency savings without error, we cannot offer a theoretical guarantee of a correct outcome under all circumstances. More evidence is required before the approach can become common practice, but BAI methods provide a principled way to estimate what is the best

Table 4 Hoeffding races

	Binary	Non-binary
Experiment I	280	380
Experiment II	190	270
Experiment III	170	210
Experiment IV	180	200
Experiment V	90	90
Experiment VI	180	280
Experiment VII	–	–

Minimum number of cases per condition that yielded some saving

condition. We note too that we treat the user study results as a gold-standard (i.e. if the same result is achieved then we judge the result to be error free). In practice, both type I and type II errors can occur in the original analyses, which we cannot account for here. This is of course not the case for the synthetic data sets where no uncertainty exists as we produced the simulation.

We have only studied a single type of user study (where the strongest condition with respect to a single metric is being sought). While, as discussed above, such studies are common in our field, other studies may instead seek to investigate the effects of different conditions on multiple dependent variables. We plan to study how to adapt BAI algorithms to such multi-purpose settings. There may also be user studies whose designs do not fit well with the BAI framework but may still benefit from another sort of adaptive device. In such cases, other MAB methods might be considered. For example, we could explore the application of MAB algorithms that handle multi-objective rewards and are oriented to maximise the overall utility (e.g., [69]).

Another point, discussed extensively in the medical literature (e.g., [73]), is the potential cost of losing randomisation (conditions are no longer randomly assigned). This is worthy of consideration as randomisation is a means to reduce, for example, learning effects and effects relating to fatigue, which could not be studied in our experiments. Further work is necessary to analyse these in detail including using approaches such as Bayesian randomisation [73]. Whereas in medicine the ethical benefits and empirical evidence for the efficacy and reliability of the approach have won the debate, it is our position that it is important for the IR community to have the debate, regardless of the outcome.

6.3 Utility in practice

One question readers may have is how they might use these results for their own studies. In practice applying the best-arm identification algorithms would mean switching participants between conditions during experiments. In cases, such as the evaluation of search algorithms, this is not a problem as it is not obvious to participants that anything has changed. For example, we could employ a BAI solution to quickly select the best retrieval method among a set of competing alternatives (e.g., multiple cluster-based methods [10, 18, 19]).

In the case of search user interfaces, this may be more problematic since dramatic interfaces changes would be obvious to participants and noticing may inherently alter their behaviour. This is something that researchers must consider when planning their studies.

To enable the changing of conditions we will make code available describing the algorithms so that experimenters can introduce them into their own pipelines. Furthermore, an online service could be developed to assist researchers in assigning conditions based on previous results. The setup would be similar in a sense to that used by NIST in the TREC CORE Track 2017 [1]. In order to generate relevance judgements, NIST utilised a MAB method [48, 49] that adaptively selected the documents to be judged by human assessors. The MAB algorithm was implemented on a server that received judgements from the assessors and returned the next suggested judgement. We could imagine setting up a similar service, where the experimenter defines the conditions and associated rewards while the algorithm drives the selection of conditions. In the case of a Crowdsourced study, the MAB algorithm could be built directly into the code and could, therefore, after initial set up, be set to run and minimise costs with no additional input or monitoring from the researcher [54].

7 Conclusions

By studying BAI algorithms using freely-available and synthetic data sets, we have presented a strong case for the utility of adaptive IR user studies. Whilst we do not wish to argue that existing approaches should be replaced, it is clear from our findings that, in the class of studies investigated, efficiency savings can be made that could lead to fewer wasted resources, more conditions being tested and less reliance on convenience samples. We hope to test these and other MAB approaches more thoroughly in the future with diverse study designs. More specifically, we want to study recent MAB proposals that lead to generalisable algorithms (e.g., the recent adaptation of the Sequential Halving algorithm that leverages variants of Thompson Sampling [5]) and see how their perform in comparison to the BAI solutions proposed here. We encourage researchers who perform user studies to make their data available, if they are so permitted.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work was funded by FEDER/Ministerio de Ciencia, Innovación y Universidades – Agencia Estatal de Investigación/ Project (RTI2018-093336-B-C21). This work has received financial support from the Consellería de Educación, Universidade e Formación Profesional (accreditation 2019-2022 ED431G-2019/04, ED431C 2018/29 , ED431C 2018/19) and the European Regional Development Fund (ERDF), which acknowledges the CiTIUS-Research Center in Intelligent Technologies of the University of Santiago de Compostela as a Research Center of the Galician University System.

Availability of data and material Code Availability The Nudge and Movie/Recipes datasets are available at https://ai.ur.de/fibc/datasets.html. The Privacy dataset is available at https://github.com/stevenzim/chiir-2019. Contact with the first author to get access to the R implementation of the BAI algorithms.

Declarations

Conflict of Interests The authors declare no conflict of interest.

References

1. Allan J, Harman D, Kanoulas E, Li D, Gysel CV, Voorhees EM (2017) TREC 2017 common core track overview. In: Proceedings of TREC '17
2. Audibert J-Y, Bubeck S, Munos R (2010) Best arm identification in multi-armed bandits. In: Proceedings of COLT '10
3. Audibert J-Y, Munos R, Szepesvári C (2007) Tuning bandit algorithms in stochastic environments. In: Proceedings of ALT '07
4. Aula A, Jhaveri N, Käki M (2005) Information search and re-access strategies of experienced web users. In: Proceedings of WWW '05
5. Aziz M, Kaufmann E, Riviere M-K (2021) On multi-armed bandit designs for dose-finding clinical trials. J Mach Learn Res 22:1–38
6. Bacchetti P (2010) Current sample size conventions: flaws, harms, and alternatives. BMC Med 8(1)
7. Bartlett RH, Roloff DW, Cornell RG, Andrews AF, Dillon PW, Zwischenberger JB (1985) Extracorporeal circulation in neonatal respiratory failure: a prospective randomized study. Pediatrics 76(4)
8. Bauer P, Kieser M (1999) Combining different phases in the development of medical treatments within a single trial. Stat Med 18(14)
9. Bendersky M, Garcia-Pueyo L, Harmsen J, Josifovski V, Lepikhin D (2014) Up next: retrieval methods for large scale related video suggestion. In: Proceedings of KDD '14
10. Bhopale AP, Tiwari A (2020) Swarm optimized cluster based framework for information retrieval. Expert Syst Appl 154:113441
11. Burtini G, Loeppky J, Lawrence R (2015) A survey of online experiment design with the stochastic multi-armed bandit. arXiv:1510.00757
12. Caine K (2016) Local standards for sample size at chi. In: Proceedings of CHI '16
13. Chandler J, Shapiro D (2016) Conducting clinical research using crowdsourced convenience samples. Ann Rev Clin Psychol 12
14. Chow S-C, Chang M (2008) Adaptive design methods in clinical trials–a review. Orphanet J Rare Dis 3(1)
15. Heting C, Qing K (2017) Research methods: What's in the name? Libr Inf Sci Res 39(4)
16. Colton T (1962) A model for selecting one of two medical treatments. Bull Inst Int Statist 39(3)
17. Dervin B, Nilan M (1986) Information needs and uses. Ann Rev Inf Sci Technol 21
18. Djenouri Y, Belhadi A, Djenouri D, Lin C-W (2021) Cluster-based information retrieval using pattern mining. Appl Intell 51:1–16
19. Djenouri Y, Belhadi A, Fournier-Viger P, Lin JC-W (2018) Fast and effective cluster-based information retrieval using frequent closed itemsets. Inform Sci 453:154–167
20. Ellis D, Haugan M (1997) Modelling the information seeking patterns of engineers and research scientists in an industrial environment. J Doc 53(4)
21. Elsweiler D, Trattner C, Harvey M (2017) Exploiting food choice biases for healthier recipe recommendation. In: Proceedings of SIGIR '17
22. Epstein S (2009) Inclusion: the politics of difference in medical research. Chicago Studies in Practices of Meaning
23. Even-Dar E, Mannor S, Mansour Y (2006) Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. J Mach Learn Res 7
24. Even-Dar E, Mannor S, Mansour Y (2012) PAC bounds for multi-armed bandit and markov decision processes. In: Proceedings of COLT '02
25. Fern EF, Monroe KB (1996) Effect-size estimates: Issues and problems in interpretation. J Consum Res 23(2)
26. Garivier A, Kaufmann E (2016) Optimal best arm identification with fixed confidence. In: Feldman V, Rakhlin A, Shamir O (eds) 29th annual conference on learning theory, volume 49 of proceedings of machine learning research. Columbia University, New York, pp 998–1027. PMLR
27. González-González AI, Dawes M, Sánchez-Mateos J, Riesgo-Fuertes R, Escortell-Mayor E, Sanz-Cuesta T, Hernandez-Fernandez T (2007) Information needs and information-seeking behavior of primary care physicians. The Annals of Family Medicine 5(4)
28. Granmo OC, Glimsdal S (2013) Accelerated bayesian learning for decentralized two-armed bandit based decision making with applications to the goore game. Appl Intell 38:479–488
29. Greenberg S, Buxton B (2008) Usability evaluation considered harmful (some of the time). In: Proceedings of CHI '08
30. Harman D (2011) Information retrieval evaluation. Synthesis Lectures on Information Concepts Retrieval, and Services 3(2)
31. Harvey M, Hauff C, Elsweiler D (2015) Learning by example: training users with high-quality query suggestions. In: Proceedings of SIGIR '15
32. Hoeffding W (1963) Probability inequalities for sums of bounded random variables. J Am Stat Assoc 58(301)
33. Hofmann K, Whiteson S, de Rijke M (2013) Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. Inf Retr 16(1)
34. Ingwersen P, Järvelin K (2006) The turn: Integration of information seeking and retrieval in context. vol 18

35. Allan J, Harman D, Kanoulas E, Li D, Van Gysel C, Voorhees EM (2017) Trec 2017 common core track overview. In: Proceedings of the 26th text retrieval conference, TREC 2017. NIST
36. Ji Y, Li Y, Bekele BN (2007) Dose-finding in phase i clinical trials based on toxicity probability intervals. Clin Trials 4(3)
37. Kalyanakrishnan S, Tewari A, Auer P, Stone P (2012) Pac subset selection in stochastic multi-armed bandits. In: Proceedings of ICML'12
38. Kam CD, Wilking JR, Zechmeister EJ (2007) Beyond the "narrow data base": Another convenience sample for experimental research. Polit Behav 29(4)
39. Kaufmann E, Cappé O, Garivier A (2016) On the complexity of best-arm identification in multi-armed bandit models. J Mach Learn Res 17(1):1–42
40. Kelly D (2009) Methods for evaluating interactive information retrieval systems with users. Found Trends Inf Retr 3(1–2)
41. Kelly D (2015) Statistical power analysis for sample size estimation in information retrieval experiments with users. In: Proceedings of ECIR '15
42. Kelly D, Gyllstrom K (2011) An examination of two delivery modes for interactive search system experiments: remote and laboratory. In: Proceedings of CHI '11
43. Knijnenburg BP (2012) Conducting user experiments in recommender systems. In: Proceedings of RecSys '12
44. Kuhlthau CC, Tama SL (2001) Information search process of lawyers: a call for'just for me'information services. J Doc 57(1)
45. Lagun D, Agichtein E (2011) Viewser: Enabling large-scale remote user studies of web search examination and interaction. In: Proceedings of SIGIR '11
46. Lattimore T, Szepesvári C (2020) Bandit algorithms. Cambridge University Press, Cambridge
47. Levitt SD, List JA (2007) What do laboratory experiments measuring social preferences reveal about the real world? J Econ Perspect, 21(2)
48. Losada DE, Parapar J, Barreiro A (2016) Feeling lucky? multi-armed bandits for ordering judgements in pooling-based evaluation. In: Proceedings of the 31st ACM symposium on applied computing, SAC '16. ACM, pp 1027–1034
49. Losada DE, Parapar J, Barreiro A (2017) Multi-armed bandits for adjudicating documents in pooling-based evaluation of information retrieval systems. Inf Process Manag 53(5):1005–1025
50. Maron O, Moore AW (1993) Hoeffding races: Accelerating model selection search for classification and function approximation. In: Proceedings of NIPS'93
51. Martín A, Fernández-Isabel A, Martín de Diego I, Beltrán M (2021) A survey for user behavior analysis based on machine learning techniques: current models and applications. Appl Intell
52. Mnih V, Szepesvári C, Audibert J-Y (2008) Empirical bernstein stopping. In: Proceedings of ICML '08
53. Moon T, Chu W, Li L, Zheng Z, Chang Y (2012) An online learning framework for refining recency search results with user click feedback. ACM Trans Inf Syst (TOIS) 30(4)
54. Morschheuser B, Hamari J, Koivisto J (2016) Gamification in crowdsourcing: a review. In: Proceedings of HICSS '16
55. Nielsen J (2006) Quantitative studies: How many users to test. Alertbox
56. Nielsen J (2007) Usability testing with 5 users is enough. Sited on http://www.useit.com/alertbox/
57. Peterson RA (2001) On the use of college students in social science research: Insights from a second-order meta-analysis. J Consum Res 28(3)
58. Radlinski F, Craswell N (2013) Optimized interleaving for online retrieval evaluation. In: Proceedings of WSDM '13
59. Radlinski F, Kleinberg R, Joachims T (2008) Learning diverse rankings with multi-armed bandits. In: Proceedings of ICML '08
60. Rahman M, Oh JC (2018) Graph bandit for diverse user coverage in online recommendation. Appl Intell 48:1979–1995
61. Robertson S (2008) On the history of evaluation in ir. J Inf Sci 34(4)
62. Sakai T (2016) Statistical power, and sample sizes significance: A systematic review of sigir and tois, 2006-2015. In: Proceedings of SIGIR '16
63. Spool J, Schroeder W (2001) Testing web sites: Five users is nowhere near enough. In: Proceedings of CHI '01 extended abstracts
64. Sverdlov O, Wong WK, Ryeznik Y et al (2014) Adaptive clinical trial designs for phase i cancer studies. Stat Surv 8
65. Tang X, Zhang C, Meng W, Wang K (2020) Joint user mention behavior modeling for mentionee recommendation. Appl Intell 50:2449–2464
66. Terayama K, Iwata H, Araki M, Okuno Y, Tsuda K (2017) Machine learning accelerates MD-based binding pose prediction between ligands and proteins. Bioinformatics 34(5):770–778
67. Terayama K, Shinobu A, Tsuda K, Takemura K, Kitao A (2019) evERdock BAI: Machine-learning-guided selection of protein-protein complex structure. J Chem Phys 151(21):215104
68. Trattner C, Jannach D (2019) Learning to recommend similar items from human judgments. User Modeling and User-Adapted Interaction
69. Wanigasekara N, Liang Y, Goh ST, Ye L, Williams JJ, Rosenblum DS (2019) Learning multi-objective rewards and user utility function in contextual bandits for personalized ranking. In: Proceedings of IJCAI '19
70. Wei L-J, Durham S (1978) The randomized play-the-winner rule in medical trials. J Am Stat Assoc 73(364)
71. Woolrych A, Cockton G (2001) Why and when five test users aren't enough. In: Proceedings of IHM-HCI '01, vol 2
72. Xu L, Zhou X, Gadiraju U (2019) Revealing the role of user moods in struggling search tasks. In: Proceedings of SIGIR '19
73. Yin G, Lam CK, Shi H (2017) Bayesian randomized clinical trials: From fixed to adaptive design. Contemp Clin Trials 59
74. Yue Y, Joachims T (2009) Interactively optimizing information retrieval systems as a dueling bandits problem. In: Proceedings of ICML '09
75. Zelen M (1969) Play the winner rule and the controlled clinical trial. J Am Stat Assoc 64(325)
76. Zimmerman S, Thorpe A, Fox C, Kruschwitz U (2019) Privacy nudging in search: Investigating potential impacts. In: Proceedings of CHIIR '19