JMLR: Workshop and Conference Proceedings vol 30 (2013) 1–29

Adaptive Crowdsourcing Algorithms for the Bandit Survey Problem

Ittai Abraham ITTAIA@MICROSOFT.COM
Microsoft Research Silicon Valley, Mountain View CA, USA.

Omar Alonso OMALONSO@MICROSOFT.COM
Microsoft Corporation, Mountain View CA, USA.

Vasilis Kandylas VAKANDYL@MICROSOFT.COM
Microsoft Corporation, Mountain View CA, USA.

Aleksandrs Slivkins SLIVKINS@MICROSOFT.COM
Microsoft Research Silicon Valley, Mountain View CA, USA.

Abstract

Very recently crowdsourcing has become the de facto platform for distributing and collecting human computation for a wide range of tasks and applications such as information retrieval, natural language processing and machine learning. Current crowdsourcing platforms have some limitations in the area of quality control. Most of the effort to ensure good quality has to be done by the experimenter who has to manage the number of workers needed to reach good results.

We propose a simple model for adaptive quality control in crowdsourced multiple-choice tasks which we call the *bandit survey problem*. This model is related to, but technically different from the well-known multi-armed bandit problem. We present several algorithms for this problem, and support them with analysis and simulations. Our approach is based in our experience conducting relevance evaluation for a large commercial search engine.

1. Introduction

In recent years there has been a surge of interest in automated methods for *crowdsourcing*: a distributed model for problem-solving and experimentation that involves broadcasting the problem or parts thereof to multiple independent, relatively inexpensive workers and aggregating their solutions. Automation and optimization of this process at a large scale allows to significantly reduce the costs associated with setting up, running, and analyzing the experiments. Crowdsourcing is finding applications across a wide range of domains in information retrieval, natural language processing and machine learning.

A typical crowdsourcing workload is partitioned into *microtasks* (also called Human Intelligence Tasks), where each microtask has a specific, simple structure and involves only a small amount of work. Each worker is presented with multiple microtasks of the same type, to save time on training. The rigidity and simplicity of the microtasks' structure ensures consistency across multiple multitasks and across multiple workers.

An important industrial application of crowdsourcing concerns web search. One specific goal in this domain is *relevance assessment*: assessing the relevance of search results. One popular task design involves presenting a microtask in the form of a query along with the results from the search engine. The worker has to answer one question about the relevance of the query to the result

set. Experiments such as these are used to evaluate the performance of a search engine, construct training sets, and discover queries which require more attention and potential algorithmic tuning.

Stopping / selection issues. The most basic experimental design issue for crowdsourcing is the *stopping issue*: determining how many workers the platform should use for a given microtask before it stops and outputs the aggregate answer. The workers in a crowdsourcing environment are not very reliable, so multiple workers are usually needed to ensure a sufficient confidence level. There is an obvious tradeoff here: using more workers naturally increases the confidence of the aggregate result but it also increases the cost and time associated with the experiment. One fairly common heuristic is to use less workers if the microtasks seem easy, and more workers if the microtasks seem hard. However, finding a sweet-spot may be challenging, especially if different microtasks have different degrees of difficulty.

Whenever one can distinguish between workers, we have a more nuanced *selection issue*: which workers to choose for a given microtask? The workers typically come from a large, loosely managed population. Accordingly, the skill levels vary over the population, and are often hard to predict in advance. Further, the relative skill levels among workers may depend significantly on a particular microtask or type of microtasks. Despite this uncertainty, it is essential to choose workers that are suitable or cost-efficient for the micro-task at hand, to the degree of granularity allowed by the crowdsourcing platform. For example, while targeting individual workers may be infeasible, one may be able to select some of the workers' attributes such as age range, gender, country, or education level. Also, the crowdsourcing platform may give access to multiple third-party providers of workers, and allow to select among those.

Our focus. This paper is concerned with a combination of the stopping / selection issues discussed above. We seek a clean setting so as to understand these issues at a more fundamental level.

We focus on the scenario where several different populations of workers are available and can be targeted by the algorithm. As explained above, these populations may correspond to different selections of workers' attributes, or to multiple available third-party providers. We will refer to such populations as *crowds*. We assume that the quality of each crowd depends on a particular microtask, and is not known in advance.

Each microtask is processed by an online algorithm which can adaptively decide which crowd to ask next. Informally, the goal is target the crowds that are most suitable for this microtask. Eventually the algorithm must stop and output the aggregate answer.

This paper focuses on processing a single microtask. This allows us to simplify the setting: we do not need to model how the latent quantities are correlated across different microtasks, and how the decisions and feedbacks for different microtasks are interleaved over time. Further, we separate the issue of learning the latent quality of a crowd for a given microtask from the issue of learning the (different but correlated) quality parameters of this crowd across multiple microtasks.

Our model: the bandit survey problem. We consider microtasks that are multiple-choice questions: one is given a set $\mathcal{O}$ of possible answers, henceforth called *options*. We allow more than two options. (In fact, we find this case to be much more difficult than the case of only two options.) Informally, the microtask has a unique correct answer $x^* \in \mathcal{O}$, and the high-level goal of the algorithm is to find it.

The algorithm has access to several crowds: populations of workers. Each crowd i is represented by a distribution $\mathcal{D}_i$ over $\mathcal{O}$, called the *response distribution* for i. We assume that all crowds agree on the correct answer:[1] some option $x^* \in \mathcal{O}$ is the unique most probable option for each $\mathcal{D}_i$.

In each round t, the algorithm picks some crowd $i = i_t$ and receives an independent sample from the corresponding response distribution $\mathcal{D}_i$. Eventually the algorithm must stop and output its guess for x^*. Each crowd i has a known per-round cost c_i. The algorithm has two objectives to minimize: the total cost $\sum_t c_{i_t}$ and the *error rate*: the probability that it makes a mistake, i.e. outputs an option other than x^*. There are several ways to trade off these two objectives; we discuss this issue in more detail later in this section.

The independent sample in the above model abstracts the following interaction between the algorithm and the platform: the platform supplies a worker from the chosen crowd, the algorithm presents the microtask to this worker, and the worker picks some option.

Alternative interpretation. The crowds can correspond not to different populations of workers but to different ways of presenting the same microtask. For example, one could vary the instructions, the order in which the options are presented, the fonts and the styles, and the accompanying images.

The name of the game. Our model is similar to the extensively studied *multi-armed bandit problem* (henceforth, *MAB*) in that in each round an algorithm selects one alternative from a fixed and known set of available alternatives, and the feedback depends on the chosen alternative. However, while an MAB algorithm collects rewards, an algorithm in our model collects a *survey* of workers' opinions. Hence we name our model the **bandit survey problem**.

Discussion of the model. The bandit survey problem belongs to a broad class of online decision problems with explore-exploit tradeoff: that is, the algorithm faces a tradeoff between collecting information (*exploration*) and taking advantage of the information gathered so far (*exploitation*). The paradigmatic problem in this class is MAB: in each round an algorithm picks one alternative (*arm*) from a given set of arms, and receives a randomized, time-dependent reward associated with this arm; the goal is to maximize the total reward over time. Most papers on explore-exploit tradeoff concern MAB and its variants.

The bandit survey problem is different from MAB in several key respects. First, the feedback is different: the feedback in MAB is the reward for the chosen alternative, whereas in our setting the feedback is the opinion of a worker from the chosen crowd. While the information received by a bandit survey algorithm can be interpreted as a "reward", the value of such reward is not revealed to the algorithm and moreover not explicitly defined. Second, the algorithm's goal is different: the goal in MAB is to maximize the total reward over time, whereas the goal in our setting is to output the correct answer. Third, in our setting there are two types of "alternatives": crowds and options in the microtask. Apart from repeatedly selecting between the crowds, a bandit survey algorithm needs to output one option: the aggregate answer for the microtask.

An interesting feature of the bandit survey problem is that an algorithm for this problem consists of two components: a *crowd-selection algorithm* – an online algorithm that decides which crowd to ask next, and a *stopping rule* which decides whether to stop in a given round and which option to output as the aggregate answer. These two components are, to a large extent, independent from one another: as long as they do not explicitly communicate with one another (or otherwise share a common communication protocol) any crowd-selection algorithm can be used in conjunction with

1. Otherwise the algorithm's high-level goal is less clear. We chose to avoid this complication in the current version.

any stopping rule.[2] The conceptual separation of a bandit survey algorithm into the two components is akin to one in Mechanism Design, where it is very useful to separate a "mechanism" into an "allocation algorithm" and a "payment rule", even though the two components are not entirely independent of one another.

Trading off the total cost and the error rate. In the bandit survey problem, an algorithm needs to trade off the two objectives: the total cost and the error rate. In a typical application, the customer is willing to tolerate a certain error rate, and wishes to minimize the total cost as long as the error rate is below this threshold. However, as the error rate depends on the problem instance, there are several ways to make this formal. Indeed, one could consider the worst-case error rate (the maximum over all problem instances), a typical error rate (the expectation over a given "typical" distribution over problem instance), or a more nuanced notion such as the maximum over a given family of "typical" distributions. Note that the "worst-case" guarantees may be overly pessimistic, whereas considering "typical" distributions makes sense only if one knows what these distributions are.

For our theoretical guarantees, we focus on the worst-case error rate, and use the *bi-criteria objective*, a standard approach from theoretical computer science literature: we allow some slack on one objective, and compare on another. In our case, we allow slack on the worst-case error rate, and compare on the expected total cost. More precisely: we consider a benchmark with some worst-case error rate $\delta > 0$ and optimal total cost given this δ, allow our algorithm to have worst-case error rate which is (slightly) larger than δ, and compare its expected total cost to that of the benchmark.

Moreover, we obtain provable guarantees in terms of a different, problem-specific objective: use the same stopping rule, compare on the expected total cost. We believe that such results are well-motivated by the structure of the problem, and provide a more informative way to compare crowd-selection algorithms.

In our experiments, we fix the per-instance error rate, and compare on the expected total cost.

An alternative objective is to assign a monetary penalty to a mistake, and optimize the overall cost, i.e. the cost of labor minus the penalty. However, it may be exceedingly difficult for a customer to assign such monetary penalty,[3] whereas it is typically feasible to specify tolerable error rates. While we think this alternative is worth studying, we chose not to follow it in this paper.

Our approach: independent design. Our approach is to design crowd-selection algorithms and stopping rules independently from one another. We make this design choice in order to make the overall algorithm design task more tractable. While this is not the only possible design choice, we find it productive, as it leads to a solid theoretical framework and algorithms that are practical and theoretically founded.

Given this "independent design" approach, one needs to define the design goals for each of the two components. These goals are not immediately obvious. Indeed, two stopping rules may compare differently depending on the problem instance and the crowd-selection algorithms they are used with. Likewise, two crowd-selection algorithms may compare differently depending on the problem instance and the stopping rules they are used with. Therefore the notions of optimal stopping rule and optimal crowd-selection algorithm are not immediately well-defined.

We resolve this conundrum as follows. We design crowd-selection algorithms that work well across a wide range of stopping rules. For a fair comparison between crowd-selection algorithms,

2. The no-communication choice is quite reasonable: in fact, it can be complicated to design a reasonable bandit survey algorithm that requires explicit communication between the crowd-selection algorithm and a stopping rule.
3. In particular, this was the case in the authors' collaboration with a commercial crowdsourcing platform.

we use them with the *same* stopping rule (see Section 3 for details), and argue that such comparison is consistent across different stopping rules.

Our contributions. We introduce the bandit survey problem and present initial results in several directions: benchmarks, algorithms, theoretical analysis, and experiments.

We are mainly concerned with the design of crowd-selection algorithms. Our crowd-selection algorithms work with arbitrary stopping rules. While we provide a specific (and quite reasonable) family of stopping rules for concreteness, third-party stopping rules can be easily plugged in.

For the theoretical analysis of crowd-selection algorithms, we use a standard benchmark: the best time-invariant policy given all the latent information. The literature on online decision problems typically studies a deterministic version of this benchmark: the best fixed alternative (in our case, the best fixed crowd). We call it the *deterministic benchmark*. We also consider a randomized version, whereby an alternative (crowd) is selected independently from the same distribution in each round; we call it the *randomized benchmark*. The technical definition of the benchmarks, as discussed in Section 3, roughly corresponds to equalizing the worst-case error rates and comparing costs.

The specific contributions are as follows.

(1) We largely solve the bandit survey problem as far as the deterministic benchmark is concerned. We design two crowd-selection algorithms, obtain strong provable guarantees, and show that they perform well in experiments.

Our provable guarantees are as follows. If our crowd-selection algorithm uses the same stopping rule as the benchmark, we match the expected total cost of the deterministic benchmark up to a small additive factor, assuming that all crowds have the same per-round costs. This result holds, essentially, for an arbitrary stopping rule. We obtain a similar, but slightly weaker result if crowds can have different per-round costs. Moreover, we can restate this as a bi-criteria result, in which we incur a small additive increase in the expected total cost and $(1 + k)$ multiplicative increase in the worst-case error rate, where k is the number of crowds. The contribution in these results is mostly conceptual rather than technical: it involves "independent design" as discussed above, and a "virtual rewards" technique which allows us to take advantage of the MAB machinery.

For comparison, we consider a naive crowd-selection algorithm that tries each crowd in a round-robin fashion. We prove that this algorithm, and more generally any crowd-selection algorithm that does not adapt to the observed workers' responses, performs very badly against the deterministic benchmark. While one expects this on an intuitive level, the corresponding mathematical statement is not easy to prove. In experiments, our proposed crowd-selection algorithms perform much better than the naive approach.

(2) We observe that the randomized benchmark dramatically outperforms the deterministic benchmark on some problem instances. This is a very unusual property for an online decision problem.[4] (However, the two benchmarks coincide when there are only two possible answers.)

We design an algorithm which significantly improves over the expected total cost of the deterministic benchmark on some problem instances (while not quite reaching the randomized benchmark), when both our algorithm and the benchmarks are run with the same stopping rule. This appears to be the first published result in the literature on online decision problems where an algorithm provably improves over the deterministic benchmark.

4. We are aware of only one published example of an online decision problem with this property, in a very different context of dynamic pricing (Babaioff et al., 2012). However, the results in (Babaioff et al., 2012) focus on a special case where the two benchmarks essentially coincide.

We can aslo restate this result in terms of the bi-criteria objective. Then we suffer a $(1+k)$ multiplicative increase in the worst-case error rate.

(3) We provide a specific stopping rule for concreteness; this stopping rule is simple, tunable, has nearly optimal theoretical guarantees (in a certain formal sense), and works well in experiments.

Preliminaries and notation. There are k crowds and n options (possible answers to the microtask). $\mathcal{O}$ denotes the set of all options. An important special case is *uniform costs*: all c_i are equal; then the total cost is simply the stopping time.

Fix round t in the execution of a bandit survey algorithm. Let $N_{i,t}$ be the number of rounds before t in which crowd i has been chosen by the algorithm. Among these rounds, let $N_{i,t}(x)$ be the number of times a given option $x \in \mathcal{O}$ has been chosen by this crowd. The *empirical distribution* $\widehat{\mathcal{D}}_{i,t}$ for crowd i is given by $\widehat{\mathcal{D}}_{i,t}(x) = N_{i,t}(x)/N_{i,t}$ for each option x. We use $\widehat{\mathcal{D}}_{i,t}$ to approximate the (latent) response distribution $\mathcal{D}_i$.

Define the *gap* $\epsilon(\mathcal{D})$ of a finite-support probability distribution $\mathcal{D}$ as the difference between the largest and the second-largest probability values in $\mathcal{D}$. If there are only two options ($n = 2$), the gap of a distribution over $\mathcal{O}$ is simply the bias towards the correct answer. Let $\epsilon_i = \epsilon(\mathcal{D}_i)$ and $\widehat{\epsilon}_{i,t} = \epsilon(\widehat{\mathcal{D}}_{i,t})$ be, respectively, the *gap* and the *empirical gap* of crowd i.

We will use vector notation over crowds: the *cost vector* $\vec{c} = (c_1, \ldots, c_k)$, the *gap vector* $\vec{\epsilon} = (\epsilon_1, \ldots, \epsilon_k)$, and the *response vector* $\vec{D}(x) = (\mathcal{D}_1(x), \ldots, \mathcal{D}_k(x))$ for each option $x \in \mathcal{O}$.

Map of the paper. The rest of the paper is organized as follows. As a warm-up and a foundation, we consider stopping rules for a single crowd (Section 2). Benchmarks are formally defined in Section 3. Design of crowd-selection algorithms with respect to the deterministic benchmark is treated in Section 4. We further discuss the randomized benchmark, and design an algorithm for it, in Section 5. We discuss open questions in Section 6.

Due to space limitations, much of the material is presented in the appendices. Related work is treated in Appendix A. Our experimental results are presented in Appendix C (for a single crowd), and Appendix D (for selection over multiple crowds). Our results in terms of the bi-criteria objective are in Appendix B.

Most proofs are moved to appendices. Out of those, the most significant ones are the lower bound for non-adaptive crowd-selection (Appendix G), and the analysis of the algorithm that competes against the randomized benchmark (Appendix I).

2. A warm-up: single-crowd stopping rules

Consider a special case with only one crowd to choose from. It is clear that whenever a bandit survey algorithm decides to stop, it should output the most frequent option in the sample. Therefore the algorithm reduces to what we call a *single-crowd stopping rule*: an online algorithm which in every round inputs an option $x \in \mathcal{O}$ and decides whether to stop. When multiple crowds are available, a single-crowd stopping rule can be applied to each crowd separately. This discussion of the single-crowd stopping rules, together with the notation and tools that we introduce along the way, forms a foundation for the rest of the paper.

A single-crowd stopping rule is characterized by two quantities that are to be minimized: the expected stopping time and the *error rate*: the probability that once the rule decides to stop, the most frequent option in the sample is not x^*. Note that both quantities depend on the problem instance.

A simple single-crowd stopping rule. We suggest the following single-crowd stopping rule:

$$\text{Stop if } \widehat{\epsilon}_{i,t}\, N_{i,t} > C_{\texttt{qty}} \sqrt{N_{i,t}}. \tag{1}$$

Here i is the crowd the stopping rule is applied to, and $C_{\texttt{qty}}$ is the *quality parameter* which indirectly controls the tradeoff between the error rate and the expected stopping time. Specifically, increasing $C_{\texttt{qty}}$ decreases the error rate and increases the expected stopping time. If there are only two options, call them x and y, then the left-hand side of the stopping rule is simply $|N_{i,t}(x) - N_{i,t}(y)|$.

The right-hand side of the stopping rule is a confidence term, which should be large enough to guarantee the desired confidence level. The $\sqrt{N_{i,t}}$ is there because the standard deviation of the Binomial distribution with N samples is proportional to $\sqrt{N}$.

In our experiments, we use a "smooth" version of this stopping rule: we randomly round the confidence term to one of the two nearest integers. In particular, the smooth version is meaningful even with $C_{\texttt{qty}} < 1$ (whereas the deterministic version with $C_{\texttt{qty}} < 1$ always stops after one round).

Analysis. We argue that the proposed single-crowd stopping rule is quite reasonable. To this end, we obtain a provable guarantee on the tradeoff between the expected stopping time and the worst-case error rate. Further, we prove that this guarantee is nearly optimal across all single-crowd stopping rules. Both results above are in terms of the gap of the crowd that the stopping rule interacts with. We conclude that the gap is a crucial parameter for the bandit survey problem.

Theorem 1 *Consider the stopping rule (1) with $C_{\texttt{qty}} = \log^{1/2}(\frac{n}{\delta}\, N_{i,t}^2)$, for some $\delta > 0$. The error rate of this stopping rule is at most $O(\delta)$, and the expected stopping time is at most $O(\ \epsilon_i^{-2}\ \log \frac{n}{\delta\epsilon_i}\ )$.*

The proof of Theorem 1, and other proofs in the paper, rely on the Azuma-Hoeffding inequality. See Appendix E for details on Azuma-Hoeffding and the proof of Theorem 1.

The following lower bound easily follows from classical results on coin-tossing. Essentially, one needs at least $\Omega(\epsilon^{-2})$ samples from a crowd with gap $\epsilon > 0$ to obtain the correct answer.

Theorem 2 *Let R_0 be any single-crowd stopping rule with worst-case error rate less than δ. When applied to a crowd with gap $\epsilon > 0$, the expected stopping time of R_0 is at least $\Omega(\epsilon^{-2} \log \frac{1}{\delta})$.*

While the upper bound in Theorem 1 is close to the lower bound in Theorem 2, it is possible that one can obtain a more efficient version of Theorem 1 using more sophisticated versions of Azuma-Hoeffding inequality such as, for example, the Empirical Bernstein Inequality.

Stopping rules for multiple crowds. For multiple crowds, we consider stopping rules that are composed of multiple instances of a given single-crowd stopping rule R_0; we call them *composite* stopping rules. Specifically, we have one instance of R_0 for each crowd (which only inputs answers from this crowd), and an additional instance of R_0 for the *total crowd* – the entire population of workers. The composite stopping rule R stops as soon as some R_0 instances stops, and outputs the majority option for this instance.[5] Given a crowd-selection algorithm $\mathcal{A}$, let $\texttt{cost}(\mathcal{A}|R_0)$ denote the expected total cost (for a given problem instance) if $\mathcal{A}$ is run together with the stopping rule R.

5. Each instance of R_0 uses an independent random seed. If multiple instances of R_0 stop at the same time, the aggregate answer is chosen uniformly at random among the majority options for the stopped instances.

3. Omniscient benchmarks for crowd selection

We consider two "omniscient" benchmarks for crowd-selection algorithms: informally, the best fixed crowd i^* and the best fixed distribution μ^* over crowds, where i^* and μ^* are chosen given the latent information: the response distributions of the crowds. Both benchmarks treat all their inputs as a single data source, and are used in conjunction with a given single-crowd stopping rule R_0 (and hence depend on the R_0).

Deterministic benchmark. Let $\texttt{cost}(i|R_0)$ be the expected total cost of always choosing crowd i, with R_0 as the stopping rule. We define the *deterministic benchmark* as the crowd i that minimizes $\texttt{cost}(i|R_0)$ for a given problem instance. In view of the analysis in Section 2, our intuition is that $\texttt{cost}(i|R_0)$ is approximated by c_i/ϵ_i^2 up to a constant factor (where the factor may depend on R_0 but not on the response distribution of the crowd). The exact identity of the best crowd may depend on R_0. For the basic special case of uniform costs and two options (assuming that the expected stopping time of R_0 is non-increasing in the gap), the best crowd is the crowd with the largest gap. In general, we approximate the best crowd by $\operatorname{argmin}_i c_i/\epsilon_i^2$.

Randomized benchmark. Given a distribution μ over crowds, let $\texttt{cost}(\mu|R_0)$ be the expected total cost of a crowd-selection algorithm that in each round chooses a crowd independently from μ, treats all inputs as a single data source – essentially, a single crowd – and uses R_0 as a stopping rule on this data source. The *randomized benchmark* is defined as the μ that minimizes $\texttt{cost}(\mu|R_0)$ for a given problem instance. This benchmark is further discussed in Section 5.

Comparison against the benchmarks. In the analysis, we compare a given crowd-selection algorithm $\mathcal{A}$ against these benchmarks as follows: we use $\mathcal{A}$ in conjunction with the composite stopping rule based on R_0, and compare the expected total cost $\texttt{cost}(\mathcal{A}|R_0)$ against those of the benchmarks.

Moreover, we derive corollaries with respect to the bi-criteria objective, where the benchmarks choose both the best crowd (resp., best distribution over crowds) and the stopping rule. These corollaries are further discussed in Appendix B.

4. Crowd selection against the deterministic benchmark

This section is on crowd-selection algorithms that compete with the deterministic benchmark.

Throughout the section, let R_0 be a fixed single-parameter stopping rule. Recall that the deterministic benchmark is defined as $\min \texttt{cost}(i|R_0)$, where the minimum is over all crowds i. We consider arbitrary composite stopping rules based on R_0, under a mild assumption that the R_0 does not favor one option over another. Formally, we assume that the probability that R_0 stops at any given round, conditional on any fixed history (sequence of observations that R_0 inputs before this round), does not change if the options are permuted. Then R_0 and the corresponding composite stopping rule are called *symmetric*. For the case of two options (when the expected stopping time of R_0 depends only on the gap of the crowd that R_0 interacts with) we sometimes make another mild assumption: that the expected stopping time decreases in the gap; we call such R_0 *gap-decreasing*.

4.1. Crowd-selection algorithms

Our crowd-selection algorithms are based on the following idea, which we call the virtual reward heuristic. For a given problem instance, consider an MAB instance where crowds correspond to arms, and selecting each crowd i results in reward $f_i = f(c_i/\epsilon_i^2)$, for some fixed decreasing function

f. (Given the discussion in Section 2, we use c_i/ϵ_i^2 as an approximation for $\texttt{cost}(i|R_0)$; we can also plug in a better approximation when and if one is available.) Call f_i the *virtual reward*; note that it is not directly observed by a bandit survey algorithm, since it depends on the gap ϵ_i. However, various off-the-shelf bandit algorithms can be restated in terms of the estimated rewards, rather than the actual observed rewards. The idea is to use such bandit algorithms and plug in our own estimates for the rewards.

A bandit algorithm thus applied would implicitly minimize the number of times suboptimal crowds are chosen. This is a desirable by-product of the design goal in MAB, which is to maximize the total (virtual) reward. (Note that we are not directly interested in this design goal.)

Algorithm 1: `UCB1` with virtual rewards. Our first crowd-selection algorithm is based on `UCB1` (Auer et al., 2002a), a standard MAB algorithm. We use virtual rewards $f_i = \epsilon_i/\sqrt{c_i}$.

We observe that `UCB1` has a property that at each time t, it only requires an estimate of f_i and a confidence term for this estimate. Motivated by Equation (6), we use $\widehat{\epsilon}_{i,t}/\sqrt{c_i}$ as the estimate for f_i, and $C/\sqrt{c_i\, N_{i,t}}$ as the confidence term. The resulting crowd-selection algorithm, called `VirtUCB`, proceeds as follows. In each round t it chooses the crowd i which maximizes the *index*

$$I_{i,t} = c_i^{-1/2}\left(\widehat{\epsilon}_{i,t} + C/\sqrt{N_{i,t}}\right). \tag{2}$$

For the analysis, we use (2) with $C = \sqrt{8\log t}$. In our experiments, $C = 1$ appears to perform best.

Algorithm 2: Thompson heuristic. Our second crowd-selection algorithm, called `VirtThompson`, is an adaptation of *Thompson heuristic* (Thompson, 1933) for MAB to virtual rewards $f_i = \epsilon_i/\sqrt{c_i}$. The algorithm proceeds as follows. For each round t and each crowd i, let $\mathcal{P}_{i,t}$ be the Bayesian posterior distribution for gap ϵ_i given the observations from crowd i up to round t (starting from the uniform prior). Sample ζ_i independently from $\mathcal{P}_{i,t}$. Pick the crowd with the largest *index* $\zeta_i/\sqrt{c_i}$. As in `UCB1`, the index of crowd i is chosen from the confidence interval for the (virtual) reward of this crowd, but here it is a random sample from this interval, whereas in `UCB1` it is the upper bound.

As it may be difficult to compute the posteriors $\mathcal{P}_{i,t}$ exactly, an approximation can be used. In our simulations we focus on the case of two options, call them x, y. For each crowd i and round t, we approximate $\mathcal{P}_{i,t}$ by the Beta distribution with shape parameters $\alpha = 1 + N_{i,t}(x)$ and $\beta = 1 + N_{i,t}(y)$, where $N_{i,t}(x) \geq N_{i,t}(y)$. (Essentially, we ignore the possibility that x is not the right answer.) It is not clear how the posterior $\mathcal{P}_{i,t}$ in our problem corresponds to the one in the original MAB problem, so we cannot directly invoke the analyses of Thompson heuristic for MAB (Chapelle and Li, 2011; Agrawal and Goyal, 2012).

A straw-man approach. In the literature on MAB, more sophisticated algorithms are often compared to the basic approach: first explore, then exploit. In our context this means to first *explore* until we can identify the best crowd, then pick this crowd and *exploit*. So for the sake of comparison we also develop a crowd-selection algorithm that is directly based on this approach; see Appendix F for the details. (This algorithm is not based on the virtual rewards.) In our experiments we find it vastly inferior to `VirtUCB` and `VirtThompson`.

4.2. Analysis: upper bounds

We obtain a lemma that captures the intuition behind the virtual reward heuristic, explaining how it helps to minimize the selection of suboptimal crowds. Then we derive an upper bound for `VirtUCB`.

Lemma 3 *Let $i^* = \operatorname{argmin}_i c_i/\epsilon_i^2$ be the approximate best crowd. Let R_0 be a symmetric single-crowd stopping rule. Then for any crowd-selection algorithm $\mathcal{A}$, letting N_i be #times crowd i is chosen, we have $\texttt{cost}(\mathcal{A}|R_0) \leq \texttt{cost}(i^*|R_0) + \sum_{i \neq i^*} c_i\, \mathbb{E}[N_i]$.*

This is a non-trivial statement because $\texttt{cost}(i^*|R_0)$ refers not to the execution of $\mathcal{A}$, but to a different execution in which crowd i^* is always chosen. The proof uses a "coupling argument".

Proof Let $\mathcal{A}^*$ be the crowd-selection algorithm which corresponds to always choosing crowd i^*.

To compare $\texttt{cost}(\mathcal{A}|R_0)$ and $\texttt{cost}(\mathcal{A}^*|R_0)$, let us assume w.l.o.g. that the two algorithms are run on correlated sources of randomness. Specifically, assume that both algorithms are run on the same realization of answers for crowd i^*: the ℓ-th time they ask this crowd, both algorithms get the same answer. Moreover, assume that the instance of R_0 that works with crowd i^* uses the same random seed for both algorithms.

Let N be the realized stopping time for $\mathcal{A}^*$. Then $\mathcal{A}$ must stop after crowd i^* is chosen N times. It follows that the difference in the realized total costs between $\mathcal{A}$ and $\mathcal{A}^*$ is at most $\sum_i c_i N_i$. The claim follows by taking expectation over the randomness in the crowds and in the stopping rule. ■

Theorem 4 (`VirtUCB`) *Let $i^* = \operatorname{argmin}_i c_i/\epsilon_i^2$ be the approximate best crowd. Let R_0 be a symmetric single-crowd stopping rule. Assume R_0 must stop after at most T rounds. Use* `VirtUCB` *with index defined by (2) with $C = \sqrt{8 \log t}$, for each round t. Let $\Lambda_i = (c_i(f_{i^*} - f_i))^{-2}$ and $\Lambda = \sum_{i \neq i^*} \Lambda_i$. Then*

$$\texttt{cost}(\texttt{VirtUCB}|R_0) \leq \texttt{cost}(i^*|R_0) + O(\Lambda \log T).$$

Proof Sketch Plugging $C = \sqrt{8 \log t}$ into Equation (6) and dividing by $\sqrt{c_i}$, we obtain the confidence bound for $|f_i - \widehat{\epsilon}_{i,t}/\sqrt{c_i}|$ that is needed in the the original analysis of `UCB1` in (Auer et al., 2002a). Then, as per that analysis, it follows that for each crowd $i \neq i^*$ and each round t we have $\mathbb{E}[N_{i,t}] \leq \Lambda_i \log t$. (This is also not difficult to derive directly.) To complete the proof, note that $t \leq T$ and invoke Lemma 3. ■

Note that the approximate best crowd i^* may be different from the (actual) best arm, so the guarantee in Theorem 4 is only as good as the difference $\texttt{cost}(i^*|R_0) - \operatorname{argmin}_i \texttt{cost}(i|R_0)$. Note that i^* is in fact the best crowd for the basic special case of uniform costs and two options (assuming that R_0 is gap-decreasing).

It is not clear whether the constants Λ_i can be significantly improved. For uniform costs we have $\Lambda_i = (\epsilon_{i^*} - \epsilon_i)^{-2}$, which is essentially the best one could hope for. This is because one needs to try each crowd $i \neq i^*$ at least $\Omega(\Lambda_i)$ times to tell it apart from crowd i^*. [6]

6. This can be proved using an easy reduction from an instance of the MAB problem where each arm i brings reward 1 with probability $(1 + \epsilon_i)/2$, and reward 0 otherwise. Treat this as an instance of the bandit survey problem, where arms correspond to crowds, and options to rewards. An algorithm that finds the crowd with a larger gap in less than $\Omega(\Lambda_i)$ steps would also find an arm with a larger expected reward, which would violate the corresponding lower bound for the MAB problem (see (Auer et al., 2002b)).

4.3. Lower bound for non-adaptive crowd selection

Consider an obvious naive approach: iterate through each crowd in a round-robin fashion. More generally, a *non-adaptive* crowd-selection algorithm is one where in each round the crowd is sampled from a fixed distribution μ over crowds. The most reasonable version, called `RandRR` (short for "randomized round-robin") is to sample each crowd i with probability $\mu_i \sim 1/c_i$.[7]

We argue that non-adaptive crowd-selection algorithms performs badly compared to `VirtUCB`. We prove that the competitive ratio of any non-adaptive crowd-selection algorithm is bounded from below by (essentially) the number of crowds. We contrast this with an upper bound on the competitive ratio of `VirtUCB`, which we derive from Theorem 4.

Here the competitive ratio of algorithm $\mathcal{A}$ (with respect to the deterministic benchmark) is defined as $\max \frac{\texttt{cost}(\mathcal{A}|R_0)}{\min_i \texttt{cost}(i|R_0)}$, where the outer $\max$ is over all problem instances in a given family of problem instances. We focus on a very simple family: problem instances with two options and uniform costs, in which one crowd has gap $\epsilon > 0$ and all other crowds have gap 0; we call such instances ϵ-*simple*. Our result holds for a version of a composite stopping rule that does not use the total crowd. Note that considering the total crowd does not, intuitively, make sense for the ϵ-simple problem instances, and we did not use it in the proof of Theorem 4, either.

Theorem 5 *Let R_0 be a symmetric single-crowd stopping rule with worst-case error rate ρ. Assume that the composite stopping rule does not use the total crowd. Consider a non-adaptive crowd-selection algorithm $\mathcal{A}$ whose distribution over crowds is μ. Then for each $\epsilon > 0$, the competitive ratio over the ϵ-simple problem instances with k crowds is at least $\frac{\sum_i c_i \mu_i}{\min_i c_i \mu_i} (1 - 2k\rho)$.*

Note that $\min \frac{\sum_i c_i \mu_i}{\min_i c_i \mu_i} = k$, where the $\min$ is taken over all distributions μ. The minimizing μ satisfies $\mu_i \sim 1/c_i$ for each crowd i, i.e. if μ corresponds to `RandRR`.

The proof of Theorem 5 is in Appendix G. Essentially, we need to compare the stopping time of the composite stopping rule R with the stopping time of the instance of R_0 that works with the gap-ϵ crowd. The main technical difficulty is to show that the other crowds are not likely to force R to stop before this R_0 instance does. The $(1 - 2k\rho)$ factor could be an artifact of our somewhat crude method to bound the "contribution" of the gap-0 crowds. We conjecture that this factor is unnecessary (perhaps under some minor assumptions on R_0).

Competitive ratio of `VirtUCB`. Consider the case of two options and uniform costs. Then (assuming R_0 is gap-decreasing) the approximate best crowd i^* in Theorem 4 is the best crowd. The competitive ratio of `VirtUCB` is, in the notation of Theorem 4, at most $1 + \frac{O(\Lambda \log T)}{\texttt{cost}(i^*|R_0)}$. This factor is close to 1 when R_0 is tuned so as to decrease the error rate at the expense of increasing the expected running time.

5. Crowd selection against the randomized benchmark

In this section we further discuss the randomized benchmark for crowd-selection algorithms, as defined in Section 3. The total crowd under a given μ behaves as a single crowd whose response distribution $\mathcal{D}_\mu$ is given by $\mathcal{D}_\mu(x) = \mathbb{E}_{i\sim\mu}[\mathcal{D}_i(x)]$ for all options x. The gap of $\mathcal{D}_\mu$ will henceforth be called the *induced gap* of μ, and denoted $f(\mu) = \epsilon(\mathcal{D}_\mu)$. If the costs are uniform then $\texttt{cost}(\mu|R_0)$

7. For uniform costs it is natural to use a uniform distribution for μ. For non-uniform costs our choice is motivated by Theorem 5, where it (approximately) minimizes the competitive ratio.

is simply the expected stopping time of R_0 on $\mathcal{D}_\mu$, which we denote $\tau(\mathcal{D}_\mu)$. Informally, $\tau(\mathcal{D}_\mu)$ is driven by the induced gap of μ.

We show that the induced gap can be much larger than the gap of any crowd.

Lemma 6 *Let μ be the uniform distribution over crowds. For any $\epsilon > 0$ there exists a problem instance such that the gap of each crowd is ϵ, and the induced gap of μ is at least $\frac{1}{10}$.*

To prove Lemma 6, consider the following problem instance: there are two crowds and three options, and the response distributions are $(\frac{2}{5}+\epsilon, \frac{2}{5}, \frac{1}{5}-\epsilon)$ and $(\frac{2}{5}+\epsilon, \frac{1}{5}-\epsilon, \frac{2}{5})$. This problem instance the induced distribution is $\mathcal{D}_\mu = (\frac{2}{5}+\epsilon, \frac{3}{10}-\frac{\epsilon}{2}, \frac{3}{10}-\frac{\epsilon}{2})$.

We conclude that the randomized benchmark does not reduce to the deterministic benchmark: in fact, it can be much stronger. Formally, this follows from Lemma 6 under a very mild assumption on R_0: that for any response distribution $\mathcal{D}$ with gap $\frac{1}{10}$ or more, and any response distribution $\mathcal{D}'$ whose gap is sufficiently small, it holds that $\tau(\mathcal{D}) \gg \tau(\mathcal{D}')$. The implication for the design of crowd-selection algorithms is that algorithms that zoom in on the best crowd may be drastically suboptimal; for some problem instances the right goal is to optimize over distributions over crowds.

However, the randomized benchmark coincides with the deterministic benchmark for some important special cases. First, the two benchmarks coincide if the costs are uniform and all crowds agree on the top *two* options (and R_0 is gap-decreasing). Second, the two benchmarks may coincide if there are only two options ($|\mathcal{O}| = 2$), see Lemma 7 below. To prove this lemma for non-uniform costs, one needs to explicitly consider $\texttt{cost}(\mu|R_0)$ rather than just argue about the induced gaps. Our proof assumes that the expected stopping time of R_0 is a concave function of the gap; it is not clear whether this assumption is necessary. The proof can be found in Appendix H.

Lemma 7 *Consider the bandit survey problem with two options ($|\mathcal{O}| = 2$). Consider a symmetric single-crowd stopping rule R_0. Assume that the expected stopping time of R_0 on response distribution $\mathcal{D}$ is a concave function of $\epsilon(\mathcal{D})$. Then the randomized benchmark coincides with the deterministic benchmark: $\texttt{cost}(\mu|R_0) \geq \min_i \texttt{cost}(i|R_0)$ for any distribution μ over crowds.*

A crowd-selection algorithm. We design a crowd-selection algorithm with guarantees against the randomized benchmark. We use (a version of) the single-crowd stopping rule R_0 from Section 2. The stopping rule is parameterized by the "quality parameter" $C_{\texttt{qty}}$ and the time horizon T. Letting $\widehat{\epsilon}_{*,t}$ be the empirical gap of the total crowd, R_0 stops upon reaching round t if and only if

$$\widehat{\epsilon}_{*,t} > C_{\texttt{qty}}/\sqrt{t} \quad \text{or} \quad t = T. \tag{3}$$

Let $\mathcal{M}$ be the set of all distributions over crowds, and let $f^* = \max_{\mu\in\mathcal{M}} f(\mu)$ be the maximal induced gap. The benchmark cost is then at least $\Omega((f^*)^{-2})$.

Theorem 8 *Consider the bandit survey problem with uniform costs. There exists a crowd-selection algorithm $\mathcal{A}$ such that $\texttt{cost}(\mathcal{A}|R_0) \leq O(\,(f^*)^{-(k+2)}\,\sqrt{\log T}\,)$.*

We interpret this guarantee as follows: we match the benchmark cost for a distribution over crowds whose induced gap is $(f^*)^{2/(k+2)}$. By Lemma 6, the gap of the best crowd may be much smaller, so this is can be a significant improvement over the deterministic benchmark. The algorithm and the analysis are discussed in detail in Appendix I.

6. Open questions

The bandit survey problem. The main open questions concern crowd-selection algorithms for the randomized benchmark. First, we do not know how to handle non-uniform costs. Second, we conjecture that our algorithm for uniform costs can be significantly improved. Moreover, it is desirable to combine guarantees against the randomized benchmark with (better) guarantees against the deterministic benchmark.

Our results prompt several other open questions. First, while we obtain strong provable guarantees for `VirtUCB`, it is desirable to extend these or similar guarantees to `VirtThompson`, since this algorithm performs best in the experiments. Second, is it possible to significantly improve over the composite stopping rules? Third, is it advantageous to forego our "independent design" approach and design the crowd-selection algorithms jointly with the stopping rules?

Extended models. It is tempting to extend our model in several directions listed below. First, while in our model the gap of each crowd does not change over time, it is natural to study settings with bounded or "adversarial" change; one could hope to take advantage of the tools developed for the corresponding versions of MAB. Second, as discussed in the introduction, an alternative model worth studying is to assign a monetary penalty to a mistake, and optimize the overall cost (i.e., cost of labor minus penalty). Third, one can combine the bandit survey problem with learning across multiple related microtasks.

Acknowledgments

We thank Ashwinkumar Badanidiyuru, Sebastien Bubeck, Chien-Ju Ho, Robert Kleinberg and Jennifer Wortman Vaughan for stimulating discussions on our problem and related research. Also, we thank Rajesh Patel, Steven Shelford and Hai Wu from Microsoft Bing for insights into the practical aspects of crowdsourcing. Finally, we are indebted to the anonymous referees for sharp comments which have substantially improved presentation. In particular, we thank anonymous reviewers for pointing out that our index-based algorithm can be interpreted via virtual rewards.

References

Shipra Agrawal and Navin Goyal. Analysis of Thompson Sampling for the multi-armed bandit problem. In *25nd Conf. on Learning Theory (COLT)*, 2012.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002a. Preliminary version in *15th ICML*, 1998.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002b. Preliminary version in *36th IEEE FOCS*, 1995.

Moshe Babaioff, Shaddin Dughmi, Robert Kleinberg, and Aleksandrs Slivkins. Dynamic pricing with limited supply. In *13th ACM Conf. on Electronic Commerce (EC)*, 2012.

R. E. Bechhofer and D. Goldsman. Truncation of the bechhofer-kiefer-sobel sequential procedure for selecting the multinomial event which has the largest probability. *Communications in Statistics Simulation and Computation*, B14:283315, 1985.

R. E. Bechhofer, S. Elmaghraby, and N. Morse. A single-sample multiple decision procedure for selecting the multinomial event which has the highest probability. *Annals of Mathematical Statistics*, 30:102119, 1959.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure Exploration in Multi-Armed Bandit Problems. *Theoretical Computer Science*, 412(19):1832–1852, 2011. Preliminary version published in *ALT 2009*.

Chris Callison-Burch. Fast, cheap, and creative: Evaluating translation quality using amazon's mechanical turk. In *ACL SIGDAT Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 286–295, 2009.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge Univ. Press, 2006.

Olivier Chapelle and Lihong Li. An Empirical Evaluation of Thompson Sampling. In *25th Advances in Neural Information Processing Systems (NIPS)*, 2011.

Xi Chen, Qihang Lin, and Dengyong Zhou. Optimistic knowledge gradient for optimal budget allocation in crowdsourcing. In *30th Intl. Conf. on Machine Learning (ICML)*, 2013.

Paul Dagum, Richard M. Karp, Michael Luby, and Sheldon M. Ross. An optimal algorithm for monte carlo estimation. *SIAM J. on Computing*, 29(5):1484–1496, 2000.

Ofer Dekel and Ohad Shamir. Vox populi: Collecting high-quality labels from a crowd. In *22nd Conf. on Learning Theory (COLT)*, 2009.

Michael J. Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. Crowddb: answering queries with crowdsourcing. In *ACM SIGMOD Intl. Conf. on Management of Data (SIGMOD)*, pages 61–72, 2011.

Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. Web-scale Bayesian click-through rate prediction for sponsored search advertising in Microsofts Bing search engine. In *27th Intl. Conf. on Machine Learning (ICML)*, pages 13–20, 2010.

Chien-Ju Ho and Jennifer Wortman Vaughan. Online task assignment in crowdsourcing markets. In *26th Conference on Artificial Intelligence (AAAI)*, 2012.

Chien-Ju Ho, Shahin Jabbari, and Jennifer Wortman Vaughan. Adaptive task assignment for crowdsourced classification. In *30th Intl. Conf. on Machine Learning (ICML)*, 2013.

J. T. Ramey Jr. and K. Alam. A sequential procedure for selecting the most probable multinomial event. *Biometrica*, 66:171–173, 1979.

Ece Kamar, Severin Hacker, and Eric Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *11th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012.

Haim Kaplan, Eyal Kushilevitz, and Yishay Mansour. Learning with attribute costs. In *37th ACM Symp. on Theory of Computing (STOC)*, pages 356–365, 2005.

David R. Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *25th Advances in Neural Information Processing Systems (NIPS)*, pages 1953–1961, 2011.

Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *18th Advances in Neural Information Processing Systems (NIPS)*, 2004.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-Armed Bandits in Metric Spaces. In *40th ACM Symp. on Theory of Computing (STOC)*, pages 681–690, 2008.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient Adaptive Allocation Rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

Edith Law and Luis von Ahn. *Human Computation*. Morgan & Claypool Publishers, 2011.

Daniel J. Lizotte, Omid Madani, and Russell Greiner. Budgeted learning of naive-bayes classifiers. In *19th Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 378–385, 2003.

Steven L.Scott. A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26:639658, 2010.

Omid Madani, Daniel J. Lizotte, and Russell Greiner. Active model selection. In *20th Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 357–365, 2004.

Shie Mannor and John N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *J. of Machine Learning Research (JMLR)*, 5:623–648, 2004. Preliminary version in *COLT*, 2003.

Volodymyr Mnih, Csaba Szepesvári, and Jean-Yves Audibert. Empirical bernstein stopping. In *25th Intl. Conf. on Machine Learning (ICML)*, pages 672–679, 2008.

Victor S. Sheng, Foster J. Provost, and Panagiotis G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *14th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 614–622, 2008.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *ACL SIGDAT Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 254–263, 2008.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285294, 1933.

Long Tran-Thanh, Matteo Venanzi, Alex Rogers, and Nicholas R. Jennings. Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks. In *12th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, 2013.

Appendix A. Related work

For general background on crowdsourcing and human computation, refer to Law and von Ahn (2011). Most of the work on crowdsourcing is usually done using platforms like *Amazon Mechanical Turk* or *CrowdFlower*. Results using those platforms have shown that majority voting is a good approach to achieve quality (Snow et al., 2008). Get Another Label (Sheng et al., 2008) explores adaptive schemes for the single-crowd case under Baysian assumptions (while our focus is on multiple-crowds and regret under non-Bayesian uncertainty). A study on machine translation quality uses preference voting for combining ranked judgments (Callison-Burch, 2009). Vox Populi (Dekel and Shamir, 2009) suggests to prune low quality workers, however their approach is not adaptive and their analysis does not provide regret bounds (while our focus is on adaptively choosing which crowds to exploit and obtaining regret bounds against an optimal algorithm that knows the quality of each crowd). Budget-Optimal Task Allocation (Karger et al., 2011) focuses on a non-adaptive solution to the task allocation problem given a prior distribution on both tasks and judges (while we focus adaptive solutions and do not assume priors on judges or tasks). From a methodology perspective, CrowdSynth focuses on addressing consensus tasks by leveraging supervised learning (Kamar et al., 2012). Adding a crowdsourcing layer as part of a computation engine is a very recent line of research. An example is CrowdDB, a system for crowdsourcing which includes human computation for processing queries (Franklin et al., 2011). CrowdDB offers basic quality control features, but we expect adoption of more advanced techniques as those systems become more available within the community.

Multi-armed bandits (MAB) have a rich literature in Statistics, Operations Research, Computer Science and Economics. A proper discussion of this literature is beyond our scope; see (Cesa-Bianchi and Lugosi, 2006) for background. Most relevant to our setting is the work on prior-free MAB with stochastic rewards: (Lai and Robbins, 1985; Auer et al., 2002a) and the follow-up work, and Thompson heuristic (Thompson, 1933). Recent work on Thompson heuristic includes (Graepel et al., 2010; L.Scott, 2010; Chapelle and Li, 2011; Agrawal and Goyal, 2012).

Our setting is superficially similar to *budgeted MAB*, a version of MAB where the goal is to find the best arm after a fixed period of exploration (e.g., (Mannor and Tsitsiklis, 2004; Bubeck et al., 2011)). Likewise, there is some similarity with the work on *budgeted active learning* (e.g. (Lizotte et al., 2003; Madani et al., 2004; Kaplan et al., 2005)), where an algorithm repeatedly chooses instances and receives correct labels for these instances, with a goal to eventually output the correct hypothesis. The difference is that in the bandit survey problem, an algorithm repeatedly chooses among *crowds*, whereas in the end the goal is to pick the correct *option*; moreover, the true "reward" or "label" for each chosen crowd is not revealed to the algorithm and is not even well-defined.

Settings similar to stopping rules for a single crowd (but with somewhat different technical objectives) were considered in prior work, e.g. Bechhofer et al. (1959), Jr. and Alam (1979), Bechhofer and Goldsman (1985), Dagum et al. (2000), Mnih et al. (2008).

In a very recent concurrent and independent work, (Ho and Vaughan, 2012; Ho et al., 2013; Chen et al., 2013; Tran-Thanh et al., 2013) studied related, but technically incomparable settings. The first three papers consider adaptive task assignment with multiple tasks and a budget constraint on the total number or total cost of the workers. In (Ho and Vaughan, 2012; Ho et al., 2013) workers arrive over time, and the algorithm selects which tasks to assign. In (Chen et al., 2013), in each round the algorithm chooses a worker and a task, and Bayesian priors are available for the difficulty of each task and the skill level of each worker (whereas our setting is prior-independent). Finally,

Tran-Thanh et al. (2013) studies a *non-adaptive* task assignment problem where the algorithm needs to distribute a given budget across multiple tasks with known per-worker costs.

Appendix B. The bi-criteria objective

In this section we state our results with respect to the bi-criteria objective, for both deterministic and randomized benchmarks. Recall that our bi-criteria objective focuses on the worst-case error rates.

We only consider the case of uniform costs. Let $k \geq 2$ be the number of crowds.

Worst-case error rates. Let R_0 be a single-crowd stopping rule. Let $\texttt{error}(R_0)$ be the worst-case error rate of R_0, taken over all single-crowd instances (i.e., all values of the gap).

Let R be the composite stopping rule based on R_0. Let $(\mathcal{A}, R_0)$ denote the bandit survey algorithm in which a crowd-selection algorithm $\mathcal{A}$ is used together with the stopping rule R. Let $\texttt{error}(\mathcal{A}|R_0)$ be the worst-case error rate of $(\mathcal{A}, R_0)$, over all problem instances. Then

$$\texttt{error}(\mathcal{A}|R_0) \leq (k+1)\,\texttt{error}(R_0). \tag{4}$$

Note that the worst-case error rate of benchmark is simply $\texttt{error}(R_0)$. (It is achieved on a problem instance in which all crowds have gap which maximizes the error rate of R_0.) Thus, using the same R_0 roughly equalizes the worst-case error rate between $\mathcal{A}$ and the benchmarks.

Absolute benchmarks. We consider benchmarks in which both the best crowd (resp., the best distribution over crowds) and the stopping rule are chosen by the benchmark. Thus, the benchmark cost is not relative to any particular single-crowd stopping rule. We call such benchmarks *absolute*.

Let $T(\rho)$ be the smallest time horizon T for which the single-crowd stopping rule in Equation (3) achieves $\texttt{error}(R_0) \leq \rho$. Fix error rate $\rho > 0$ and time horizon $T \geq T(\rho)$. We focus on symmetric, gap-decreasing single-crowd stopping rules R_0 such that $\texttt{error}(R_0) \leq \rho$ and R_0 must stop after T rounds; let $\mathcal{R}(\rho, T)$ be the family of all such stopping rules.

Fix a problem instance. Let i^* be the crowd with the largest bias, and let μ^* be the distribution over crowds with the largest induced bias. The *absolute deterministic benchmark* (with error rate ρ and time horizon $T \geq T(\rho)$) is defined as

$$\texttt{bench}(i^*, \rho, T) = \min_{R_0 \in \mathcal{R}(\rho,T)} \texttt{cost}(i^*|R_0).$$

Likewise, the *absolute randomized benchmark* is defined as

$$\texttt{bench}(\mu^*, \rho, T) = \min_{R_0 \in \mathcal{R}(\rho,T)} \texttt{cost}(\mu^*|R_0).$$

Theorem 9 (bi-criteria results) *Consider the bandit survey problem with k crowds and uniform costs. Fix error rate $\rho > 0$ and time horizon $T \geq T(\rho)$. Then:*

(a) Deterministic benchmark. *There exists a bandit survey algorithm $(\mathcal{A}, R_0)$ such that*

$$\texttt{cost}(\mathcal{A}|R_0) \leq \texttt{bench}(i^*, \rho, T) + O(\Lambda \log T), \text{ where } \Lambda = \textstyle\sum_{i \neq i^*} (\epsilon_{i^*} - \epsilon_i)^{-2},$$
$$\texttt{error}(\mathcal{A}|R_0) \leq (k+1)\,\rho.$$

(b) Randomized benchmark. *There exists a bandit survey algorithm* $(\mathcal{A}, R_0)$ *such that*

$$\begin{aligned}\texttt{cost}(\mathcal{A}|R_0) &\leq O(\log T \log \tfrac{1}{\rho})\ (\texttt{bench}(\mu^*, \rho, T))^{1+k/2}\\ \texttt{error}(\mathcal{A}|R_0) &\leq (k+1)\,\rho.\end{aligned}$$

Proof Sketch For part (a), we use the version of `VirtUCB` as in Theorem 4, with the single-crowd stopping rule R_0 from the absolute deterministic benchmark. The upper bound on $\texttt{cost}(\mathcal{A}|R_0)$ follows from Theorem 4. The upper bound on $\texttt{error}(\mathcal{A}|R_0)$ follows from Equation (4).

For part (b), we use the algorithm from Theorem 8, together with the stopping rule given by Equation (3). The stopping rule has time horizon T; the quality parameter $C_{\texttt{qty}}$ is tuned so that the worst-case error rate matches that in the absolute randomized benchmark. The upper bound on $\texttt{cost}(\mathcal{A}|R_0)$ follows from Theorem 8, and the upper and lower bounds in Section 2. The upper bound on $\texttt{error}(\mathcal{A}|R_0)$ follows from Equation (4). ■

A lower bound on the error rate. Fix a single-crowd stopping rule R_0 with $\rho = \texttt{error}(R_0)$, and a crowd-selection algorithm $\mathcal{A}$. To complement Equation (4), we conjecture that $\texttt{error}(\mathcal{A}|R_0) \geq \rho$. We prove a slightly weaker result: essentially, if the composite stopping rule does not use the total crowd, then $\texttt{error}(\mathcal{A}|R_0) \geq \rho\,(1-2k\rho)$.

We will need a mild assumption on $\mathcal{A}$: essentially, that it never commits to stop using any given crowd. Formally, $\mathcal{A}$ is called *non-committing* if for every problem instance, each time t, and every crowd i, it will choose crowd i at some time after t with probability one. (Here we consider a run of $\mathcal{A}$ that continues indefinitely, without being stopped by the stopping rule.)

Lemma 10 *Let R_0 be a symmetric single-crowd stopping rule with worst-case error rate ρ. Let $\mathcal{A}$ be a non-committing crowd-selection algorithm, and let R be the composite stopping rule based on R_0 which does not use the total crowd. If $\mathcal{A}$ is used in conjunction with R, the worst-case error rate is at least $\rho\,(1-2k\rho)$, where k is the number of crowds.*

Proof Suppose R_0 attains the worst-case error rate for a crowd with gap ϵ. Consider the problem instance in which one crowd (say, crowd 1) has gap ϵ and all other crowds have gap 0. Let $R_{(i)}$ be the instance of R_0 that takes inputs from crowd i, for each i. Let E be the event that each $R_{(i)}$, $i>1$ does not ever stop. Let E' be the event that $R_{(1)}$ stops and makes a mistake. These two events are independent, so the error rate of R is at least $\Pr[E]\ \Pr[E']$. By the choice of the problem instance, $\Pr[E'] = \rho$. And by Lemma 13, $\Pr[E] \geq 1-2k\rho$. It follows that the error rate of R is at least $\rho\,(1-2k\rho)$. ■

Appendix C. Experimental results: single crowd

We conduct two experiments. First, we analyze real-life workloads to find which gaps are typical for response distributions that arise in practice. Second, to study the performance of the single-crowd stopping rule suggested in Section 2, using a large-scale simulation with a realistic distribution of gaps. We are mainly interested in the tradeoff between the error rate and the expected stopping time. We find that this tradeoff is acceptable in practice.

Typical gaps in real-life workloads. We analyze several batches of microtasks extracted from a commercial crowdsourcing platform (approx. 3000 microtasks total). Each batch consists of microtasks of the same type, with the same instructions for the workers. Most microtasks are related to relevance assessments for a web search engine. Each microtask was given to at least 50 judges coming from the same "crowd".

In every batch, the empirical gaps of the microtasks are very close to being *uniformly distributed* over the range. A practical take-away is that assuming a Bayesian prior on the gap would not be very helpful, which justifies and motivates our modeling choice not to assume Bayesian priors. In Figure 1, we provide CDF plots for two of the batches; the plots for the other batches are similar.

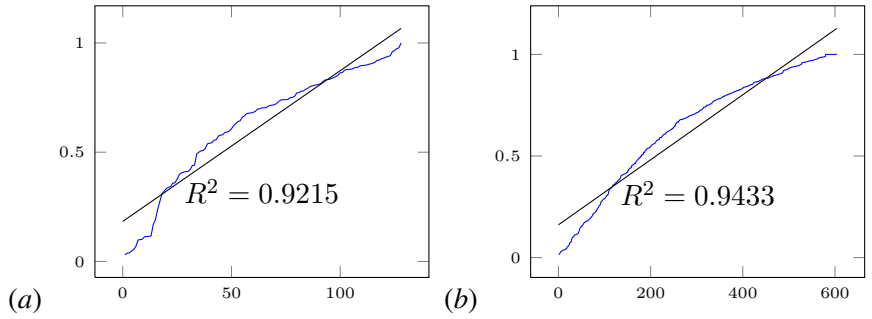


Figure 1: CDF for the empirical gap in real-life workloads.
Sub figure (a): 128 microtasks, 2 options each.
Sub figure (b): 604 microtasks, variable #options.

Our single-crowd stopping rule on simulated workloads. We study the performance of the single-crowd stopping rule suggested in Section 2. Our simulated workload consists of 10,000 microtasks with two options each. For each microtask, the gap is is chosen independently and uniformly at random in the range $[0.05, 1]$. This distribution of gaps is realistic according to the previous experiment. (Since there are only two options the gap fully describes the response distribution.)

We vary the parameter $C_{\texttt{qty}}$ and for each $C_{\texttt{qty}}$ we measure the average total cost (i.e., the stopping time averaged over all microtasks) and the error rate. The results are reported in Figure 2. In particular, for this workload, an error rate of $< 5\%$ can be obtained with an average of < 8 workers per microtask.

Our stopping rule adapts to the gap of the microtask: it uses only a few workers for easy microtasks (ones with a large gap), and more workers for harder microtasks (those with a small gap). In particular, we find that our stopping rule requires significantly smaller number of workers than a non-adaptive stopping rule: one that always uses the same number of workers while ensuring a desired error rate.

Appendix D. Experimental results: crowd-selection algorithms

We study the experimental performance of the various crowd-selection algorithms discussed in Section 4. Specifically, we consider algorithms `VirtUCB` and `VirtThompson`, and compare them to our straw-man solutions: `ExploreExploitRollback` and `RandRR`.[8] Our goal is both to com-

8. In the plots, we use shorter names for the algorithms: respectively, `VR UCB`, `VR Thompson`, `EER`, and `RR`.

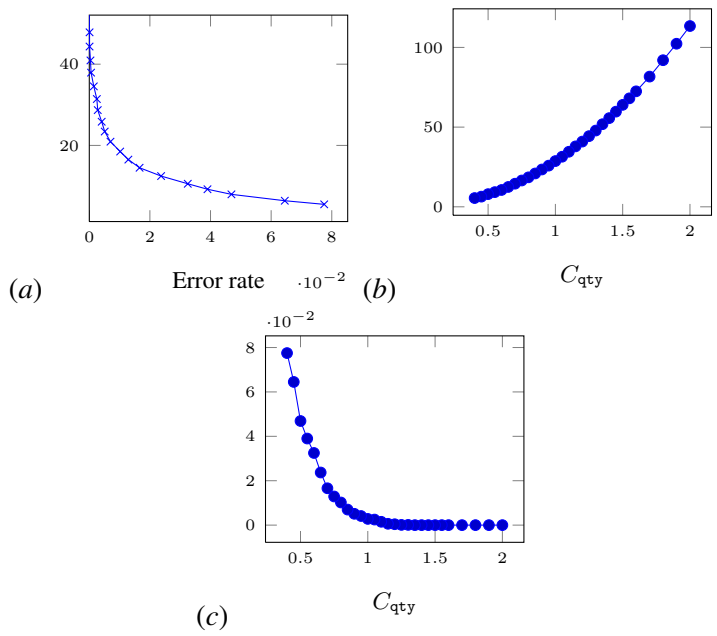


Figure 2: Our single-crowd stopping rule on the synthetic workload.
Sub figure (a): Average cost vs. error rate.
Sub figure (b): Average cost vs. $C_{\texttt{qty}}$.
Sub figure (c): Average error rate vs. $C_{\texttt{qty}}$.

pare the different algorithms and to show that the associated costs are practical. We find that `ExploreExploitRollback` consistently outperforms `RandRR` for very small error rates, `VirtUCB` significantly outperforms both across all error rates, and `VirtThompson` significantly outperforms all three.

We use all crowd-selection algorithms in conjunction with the composite stopping rule based on the single-crowd stopping rule proposed Section 2. Recall that the stopping rule has a "quality parameter" $C_{\texttt{qty}}$ which implicitly controls the tradeoff between the error rate and the expected stopping time.

We use three simulated workloads. All three workloads consist of microtasks with two options, three crowds, and unit costs. In the first workload, which we call the *easy workload*, the crowds have gaps $(0.3, 0, 0)$. That is, one crowd has gap 0.3 (so it returns the correct answer with probability 0.8), and the remaining two crowds have gap 0 (so they provide no useful information). This is a relatively easy workload for our crowd-selection algorithms because the best crowd has a much larger gap than the other crowds, which makes the best crowd easier to identify. In the second workload, called the *medium workload*, crowds have gaps $(0.3, 0.1, 0.1)$, and in the third workload, called the *hard workload*, the crowds have gaps $(0.3, 0.2, 0.2)$. The third workload is hard(er) for the crowd-selection algorithms in the sense that the best crowd is hard(er) to identify, because its gap is not much larger than the gap of the other crowds. The order that the crowds are presented to the algorithms is randomized for each instance, but is kept the same across the different algorithms.

The quality of an algorithm is measured by the tradeoff between its average total cost and its error rate. To study this tradeoff, we vary the quality parameter $C_{\texttt{qty}}$ to obtain (essentially) any desired error rate. We compare the different algorithms by reporting the average total cost of each algorithm (over 20,000 runs with the same quality parameter) for a range of error rates. Specifically, for each error rate we report the average cost of each algorithm normalized to the average cost of the naive algorithm `RandRR` (for the same error rate). See Figure 3 for the main plot: the average cost vs. error rate plots for all three workloads. Additional results, reported in Figure 4 (see page 22) show the raw average total costs and error rates for the range of values of the quality parameter $C_{\texttt{qty}}$.

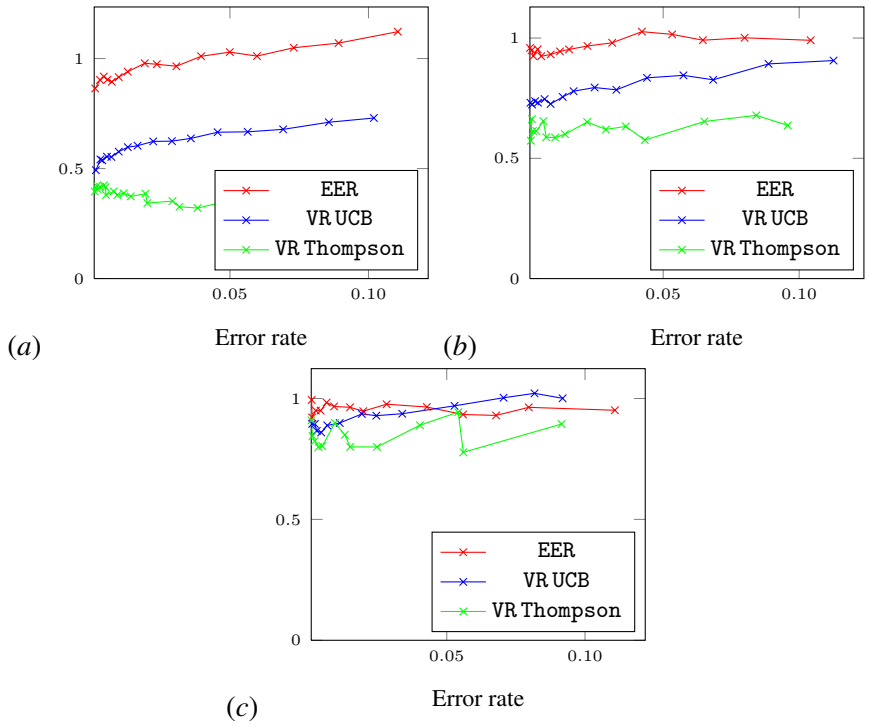
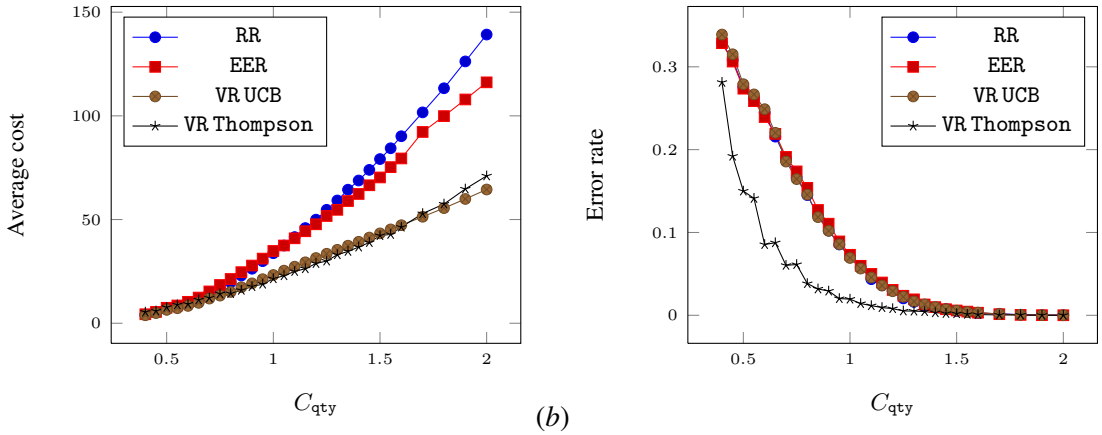


Figure 3: Crowd-selection algorithms: error rate vs. average total cost (relative to `RandRR`).
Sub-figure (a): Easy: gaps $(.3, 0, 0)$.
Sub-figure (b): Medium: gaps $(.3, .1, .1)$.
Sub-figure (c): Hard: gaps $(.3, .2, .2)$.

For `VirtUCB` we tested different parameter values for the parameter C which balances between exploration and exploitation. We obtained the best results for a range of workloads for $C = 1$ and this is the value we use in all the experiments. For `VirtThompson` we start with a uniform prior on each crowd.

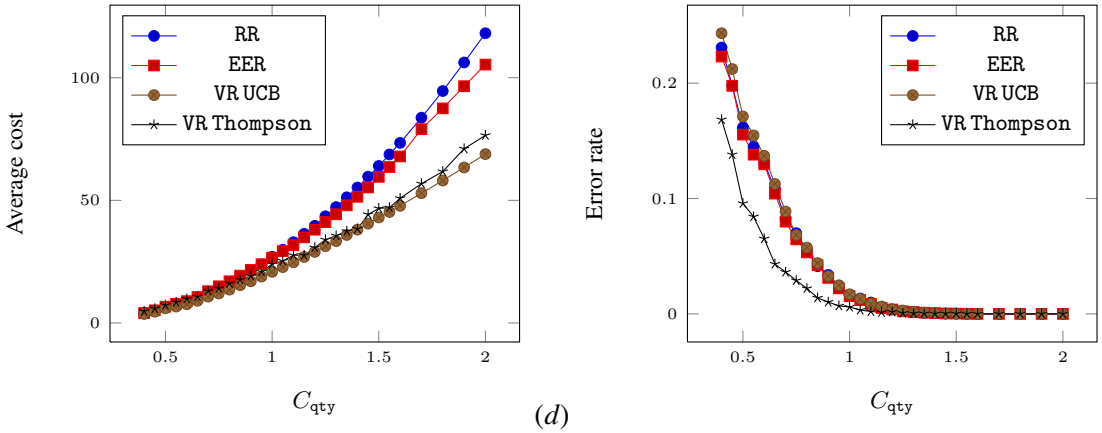
Results and discussion. For the easy workload the cost of `VirtUCB` is about 60% to 70% of the cost of `RandRR`. `VirtThompson` is significantly better, with a cost of about 40% the cost of `RandRR`. For the medium workload the cost of `VirtUCB` is about 80% to 90% of the cost of `RandRR`. `VirtThompson` is significantly better, with a cost of about 70% the cost of `RandRR`. For the hard workload the cost of `VirtUCB` is about 90% to 100% of the cost of `RandRR`. `VirtThompson` is

Additional plots for crowd-selection algorithms



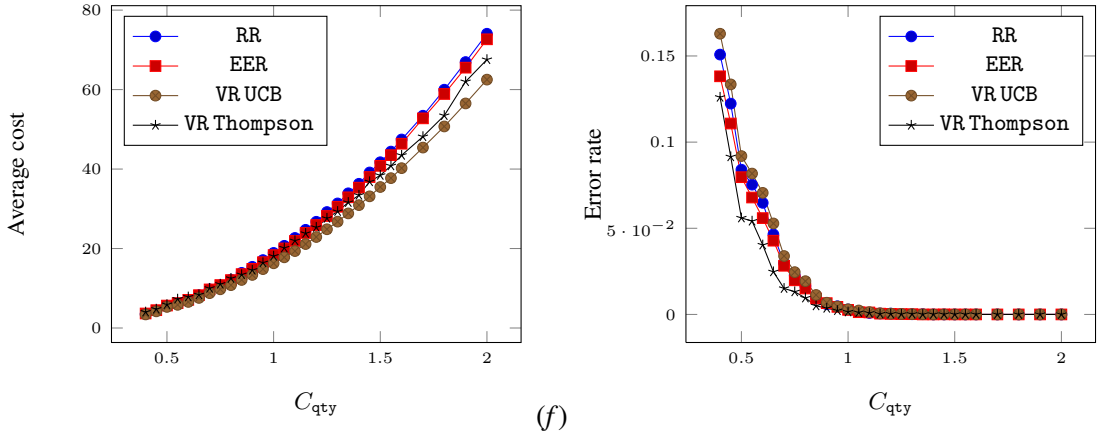
(*a*) (*b*)

The easy workload: gaps $(.3, 0, 0)$. (a) Average cost vs. $C_{\texttt{qty}}$. (b) Error rate vs. $C_{\texttt{qty}}$.



(*c*) (*d*)

The medium workload: gaps $(.3, .1, .1)$. (c) Average cost vs. $C_{\texttt{qty}}$. (d) Error rate vs. $C_{\texttt{qty}}$.



(*e*) (*f*)

The hard workload: gaps $(.3, .2, .2)$. (e) Average cost vs. $C_{\texttt{qty}}$. (f) Error rate vs. $C_{\texttt{qty}}$.

Figure 4: Crowd-selection algorithms: Average cost and error rate vs. $C_{\texttt{qty}}$.

better, with a cost of about 80% to 90% the cost of `RandRR`. While our analysis predicts that `ExploreExploitRollback` should be (somewhat) better than `RandRR`, our experiments do not confirm this for every error rate.

As the gap of the other crowds approaches that of the best crowd, choosing the best crowd becomes less important, and so the advantage of the adaptive algorithms over `RandRR` diminishes. In the extreme case where all crowds have the same gap all the algorithms would perform the same with an error rate that depends on the stopping rule. We conclude that `VirtUCB` provides an advantage, and `VirtThompson` provides a significant advantage, over the naive scheme of `RandRR`.

Appendix E. Azuma-Hoeffsing and the single-crowd stopping rule

The proof of Theorem 1, and other proofs in the paper, rely on the Azuma-Hoeffding inequality. Specifically, we use the following corollary: for each $C > 0$, each round t, and each option $x \in \mathcal{O}$

$$\Pr[\ |\mathcal{D}_i(x) - \widehat{\mathcal{D}}_{i,t}(x)| \leq C/\sqrt{N_{i,t}}\ ] \geq 1 - e^{-\Omega(C^2)}. \tag{5}$$

In particular, taking the Union Bound over all options $x \in \mathcal{O}$, we obtain:

$$\Pr[\ |\widehat{\epsilon}_{i,t} - \epsilon_i| \leq C/\sqrt{N_{i,t}}\ ] \geq 1 - n\, e^{-\Omega(C^2)}, \tag{6}$$

where n is the number of options.

Let us use Azuma-Hoeffding to prove Theorem 1. We restate the theorem here for convenience.

Theorem 11 *Consider the stopping rule (1) with $C_{\texttt{qty}} = \log^{1/2}(\frac{n}{\delta}\, N_{i,t}^2)$, for some $\delta > 0$. The error rate of this stopping rule is at most $O(\delta)$, and the expected stopping time is at most $O(\ \epsilon_i^{-2}\ \log \frac{n}{\delta\epsilon_i}\ )$.*

Proof Fix $a \geq 1$ and let $C_t = \sqrt{\log(a\, \frac{n}{\delta}\, N_{i,t}^2)}$. Let $\mathcal{E}_{x,t}$ be the event in Equation (5) with $C = C_t$. Consider the event that $\mathcal{E}_{x,t}$ holds for all options $x \in \mathcal{O}$ and all rounds t; call it the *clean event*. Taking the Union Bound, we see that the clean event holds with probability at least $1 - O(\delta/a)$.

First, assuming the clean event we have $|\epsilon_i - \widehat{\epsilon}_{i,t}| \leq 2\, C_t/\sqrt{N_{i,t}}$ for all rounds t. Then the stopping rule (1) stops as soon as $\epsilon_i \geq 3\, C_t/\sqrt{N_{i,t}}$, which happens as soon as $N_{i,t} = O\left(\epsilon_i^{-2}\ \log \frac{an}{\delta\epsilon_i}\right)$. Integrating this over all $a \geq 1$, we derive that the expected stopping time is as claimed.

Second, take $a = 1$ and assume the clean event. Suppose the stopping rule stops at some round t. Let x be the most probable option after this round. Then $\widehat{\mathcal{D}}_{i,t}(x) - \widehat{\mathcal{D}}_{i,t}(y) \geq C_t/\sqrt{N_{i,t}}$ for all options $y \neq x$. It follows that $D_i(x) > D_i(y)$ for all options $y \neq x$, i.e. x is the correct answer. ■

Appendix F. Deterministic benchmark: a straw-man approach

In the literature on MAB, more sophisticated algorithms are often compared to the basic approach: first explore, then exploit. In our context this means to first *explore* until we can identify the best crowd, then pick this crowd and *exploit*. So for the sake of comparison we also develop a crowd-selection algorithm that is directly based on this approach. (This algorithm is not based on the virtual rewards.) In our experiments we find it vastly inferior to `VirtUCB` and `VirtThompson`.

The "explore, then exploit" design does not quite work as is: selecting the best crowd with high probability seems to require a high-probability guarantee that this crowd can produce the correct answer with the current data, in which case there is no need for a further exploitation phase (and so we are essentially back to `RandRR`). Instead, our algorithm explores until it can identify the best crowd with *low* confidence, then it exploits with this crowd until it sufficiently boosts the confidence or until it realizes that it has selected a wrong crowd to exploit. The latter possibility necessitates a third phase, called *rollback*, in which the algorithm explores until it finds the right answer with high confidence.

The algorithm assumes that the single-crowd stopping rule R_0 has a quality parameter $C_{\texttt{qty}}$ which controls the trade-off between the error rate and the expected running time (as in Section 2). In the exploration phase, we also use a *low-confidence* version of R_0 that is parameterized with a lower value $C'_{\texttt{qty}} < C_{\texttt{qty}}$; we run one low-confidence instance of R_0 for each crowd.

The algorithm, called `ExploreExploitRollback`, proceeds in three phases (and stops whenever the composite stopping rule decides so). In the exploration phase, it runs `RandRR` until the low-confidence version of R_0 stops for some crowd i^*. In the exploitation phase, it always chooses crowd i^*. This phase lasts α times as long as the exploration phase, where the parameter α is chosen so that crowd i^* produces a high-confidence answer w.h.p. if it is indeed the best crowd.[9] Finally, in the roll-back phase it runs `RandRR`.

Appendix G. Lower bound for non-adaptive crowd selection (proof of Theorem 5)

We argue that non-adaptive crowd-selection algorithms performs badly compared to `VirtUCB`. We prove that the competitive ratio of any non-adaptive crowd-selection algorithm is bounded from below by (essentially) the number of crowds. This result is captured as Theorem 5 in Section 4.3, which we restate here for for convenience.

Theorem 12 (Theorem 5, restated) *Let R_0 be a symmetric single-crowd stopping rule with worst-case error rate ρ. Assume that the composite stopping rule does not use the total crowd. Consider a non-adaptive crowd-selection algorithm $\mathcal{A}$ whose distribution over crowds is μ. Then for each $\epsilon > 0$, the competitive ratio over ϵ-simple problem instances with k crowds is at least $\frac{\sum_i c_i \mu_i}{\min_i c_i \mu_i} (1-2k\rho)$.*

To prove Theorem 5, we essentially need to compare the stopping time of the composite stopping rule R with the stopping time of the instance of R_0 that works with the gap-ϵ crowd. The main technical difficulty is to show that the other crowds are not likely to force R to stop before this R_0 instance does. To this end, we use a lemma that R_0 is not likely to stop in finite time when applied to a gap-0 crowd.

Lemma 13 *Consider a symmetric single-crowd stopping rule R_0 with worst-case error rate ρ. Suppose R_0 is applied to a crowd with gap 0. Then $\Pr[R_0$ stops in finite time$] \leq 2\rho$.*

Proof Intuitively, if R_0 stops early if the gap is 0 then it is likely to make a mistake if the gap is very small but positive. However, connecting the probability in question with the error rate of R_0 requires some work.

Suppose R_0 is applied to a crowd with gap ϵ. Let $q(\epsilon, t, x)$ be the probability that R_0 stops at round t and "outputs" option x (in the sense that by the time R_0 stops, x is the majority vote).

9. We conjecture that for R_0 from Section 2 one can take $\alpha = \Theta(C_{\texttt{qty}}/C'_{\texttt{qty}})$.

We claim that for all rounds t and each option x we have

$$\lim_{\epsilon \to 0} \; q(\epsilon, t, x) = q(0, t, x). \tag{7}$$

Indeed, suppose not. Then for some $\delta > 0$ there exist arbitrarily small gaps $\epsilon > 0$ such that $|q(\epsilon, t, x) - q(0, t, x)| > \delta$. Thus it is possible to tell apart a crowd with gap 0 from a crowd with gap ϵ by observing $\Theta(\delta^{-2})$ independent runs of R_0, where each run continues for t steps. In other words, it is possible to tell apart a fair coin from a gap-ϵ coin using $\Theta(t\,\delta^{-2})$ "coin tosses", for fixed t and $\delta > 0$ and an arbitrarily small ϵ. Contradiction. Claim proved.

Let x and y be the two options, and let x be the correct answer. Let $q(\epsilon, t)$ be the probability that R_0 stops at round t. Let $\alpha(\epsilon|t) = q(\epsilon, t, y)/q(\epsilon, t)$ be the conditional probability that R_0 outputs a wrong answer given that it stops at round t. Note that by Equation (7) for each round t it holds that $q(\epsilon, t) \to q(0, t)$ and $\alpha(\epsilon|t) \to \alpha(0|t)$ as $\epsilon \to 0$. Therefore for each round $t_0 \in \mathbb{N}$ we have:

$$\rho = \textstyle\sum_{t\in\mathbb{N}} \; \alpha(\epsilon|t)\; q(\epsilon, t) \geq \sum_{t\leq t_0} \; \alpha(\epsilon|t)\; q(\epsilon, t) \to_{\epsilon\to\infty} \sum_{t\leq t_0} \; \alpha(0|t)\; q(0, t).$$

Note that $\alpha(0|t) = \frac{1}{2}$ by symmetry. It follows that $\sum_{t\leq t_0} \; q(0, t) \leq 2\rho$ for each $t_0 \in \mathbb{N}$. Therefore the probability that R_0 stops in finite time is $\sum_{t=1}^{\infty} \; q(0, t) \leq 2\rho$. ■

Proof of Theorem 5 Suppose algorithm $\mathcal{A}$ is applied to an ϵ-simple instance of the bandit survey problem. To simplify the notation, assume that crowd 1 is the crowd with gap ϵ (and all other crowds have gap 0).

Let $R_{(i)}$ be the instance of R_0 that corresponds to a given crowd i. Denote the composite stopping rule by R. Let σ_R be the stopping time of R: the round in which R stops.

For the following two definitions, let us consider an execution of algorithm $\mathcal{A}$ that runs forever (i.e., it keeps running even after R decides to stop). First, let τ_i be the "local" stopping time of $R_{(i)}$: the number of samples from crowd i that $R_{(i)}$ inputs before it decides to stop. Second, let σ_i be the "global" stopping time of $R_{(i)}$: the round when $R_{(i)}$ decides to stop. Note that $\sigma_R = \min_i \; \sigma_i$.

Let us use Lemma 13 to show that R stops essentially when $R_{(1)}$ tells it to stop. Namely:

$$\mathbb{E}[\sigma_1]\; (1 - 2k\rho) \leq \mathbb{E}[\sigma_R]. \tag{8}$$

To prove Equation (8), consider the event $E \triangleq \{\min_{i>1} \tau_i = \infty\}$, and let 1_E be the indicator variable of this event. Note that $\sigma_R \geq \sigma_1\, 1_E$ and that random variables σ_1 and 1_E are independent. It follows that $\mathbb{E}[\sigma_R] \geq \Pr[E]\, \mathbb{E}[\sigma_1]$. Finally, Lemma 13 implies that $\Pr[E] \geq 1 - 2k\rho$. Claim proved.

Let i_t be the option chosen by $\mathcal{A}$ in round t. Then by Wald's identity we have

$$\mathbb{E}[\tau_1] = \mathbb{E}\left[\sum_{t=1}^{\sigma_1} 1_{\{i_t=1\}}\right] = \mathbb{E}[1_{\{i_t=1\}}]\; \mathbb{E}[\sigma_1] = \mu_1\; \mathbb{E}[\sigma_1]$$

$$\mathbb{E}[\texttt{cost}(\mathcal{A}|R_0)] = \mathbb{E}\left[\sum_{t=1}^{\sigma_R} c_{i_t}\right] = \mathbb{E}[c_{i_t}]\; \mathbb{E}[\sigma_R] = (\textstyle\sum_i c_i\, \mu_i)\; \mathbb{E}[\sigma_R].$$

Therefore, plugging in Equation (8), we obtain

$$\frac{\mathbb{E}[\texttt{cost}(\mathcal{A}|R_0)]}{c_1\; \mathbb{E}[\tau_1]} \geq \frac{\sum_i c_i\, \mu_i}{c_1\, \mu_1}\; (1 - 2k\rho).$$

It remains to observe that $c_1\,\mathbb{E}[\tau_1]$ is precisely the expected total cost of the deterministic benchmark. ■

Appendix H. Benchmark comparison: proof of Lemma 7 from Section 5

We prove that the randomized benchmark may coincide with the deterministic benchmark if there are only two options ($|\mathcal{O}| = 2$). This result is captured as Lemma 7 in Section 5. We restate this lemma for the sake of convenience.

Lemma 14 (Lemma 7, restated) *Consider the bandit survey problem with two options ($|\mathcal{O}| = 2$). Consider a symmetric single-crowd stopping rule R_0. Assume that the expected stopping time of R_0 on response distribution $\mathcal{D}$ is a concave function of $\epsilon(\mathcal{D})$. Then the randomized benchmark coincides with the deterministic benchmark: $\texttt{cost}(\mu|R_0) \geq \min_i \texttt{cost}(i|R_0)$ for any distribution μ over crowds.*

Proof Let μ be an arbitrary distribution over crowds. Recall that $f(\mu)$ denotes the induced gap of μ. Note that $f(\mu) = \mu \cdot \vec{\epsilon}$. To see this, let $\mathcal{O} = \{x, y\}$, where x is the correct answer, and write

$$\epsilon(\mathcal{D}_\mu) = \mathcal{D}_\mu(x) - \mathcal{D}_\mu(y) = \mu \cdot \vec{D}(x) - \mu \cdot \vec{D}(y) = \mu \cdot \Big(\vec{D}(x) - \vec{D}(y)\Big) = \mu \cdot \vec{\epsilon}.$$

Let $\mathcal{A}$ be the non-adaptive crowd-selection algorithm that corresponds to μ. For each round t, let i_t be the crowd chosen by $\mathcal{A}$ in this round, i.e. an independent sample from μ. Let N be the realized stopping time of $\mathcal{A}$. Let $\tau(\epsilon)$ be the expected stopping time of R_0 on response distribution with gap ϵ. Note that $\mathbb{E}[N] = \tau(f(\mu))$. Therefore:

$$\begin{aligned}
\texttt{cost}(\mu|R_0) &= \mathbb{E}\left[\textstyle\sum_{t=1}^{N} c_{i_t}\right] = \mathbb{E}[c_{i_t}]\;\mathbb{E}[N] && \text{by Wald's identity}\\
&= (\vec{c}\cdot\mu)\;\; \tau(\vec{\epsilon}\cdot\mu) \geq (\vec{c}\cdot\mu)\;\textstyle\sum_i\;\mu_i\,\tau(\epsilon_i) && \text{by concavity of } \tau(\cdot)\\
&\geq \min_i\;\; c_i\,\tau(\epsilon_i) = \min_i \texttt{cost}(i|R_0).
\end{aligned}$$

We have used a general fact that $(\vec{x}\cdot\vec{\alpha})(\vec{x}\cdot\vec{\beta}) \geq \min_i \alpha_i\beta_i$ for any vectors $\vec{\alpha}, \vec{\beta} \in \mathbb{R}^k_+$ and any k-dimensional distribution $\vec{x}$. See Claim 15 below. ■

Claim 15 $(\vec{x}\cdot\vec{\alpha})(\vec{x}\cdot\vec{\beta}) \geq \min_i \alpha_i\beta_i$ *for any $\vec{\alpha}, \vec{\beta} \in \mathbb{R}^k_+$ and k-dimensional distribution $\vec{x}$.*

This inequality appears standard, although we have not been able to find a reference. We supply is a self-contained proof below.

Proof W.l.o.g. assume $\alpha_1\beta_1 \leq \alpha_2\beta_2 \leq \ldots \leq \alpha_k\beta_k$. Let us use induction on k, as follows. Let

$$f(\vec{x}) \triangleq (\vec{x}\cdot\vec{\alpha})(\vec{x}\cdot\vec{\beta}) = (x_1\alpha_1 + A)(x_1\beta_1 + B)$$

where

$$\begin{cases} A &= \sum_{i>1} x_i\alpha_i \\ B &= \sum_{i>1} x_i\beta_i \end{cases}.$$

Denoting $p = x_1$, we can write the above expression as

$$f(\vec{x}) = p^2\alpha_1\beta_1 + p(\alpha_1 B + \beta_1 A) + AB. \tag{9}$$

First, let us invoke the inductive hypothesis to handle the AB term in Equation (9). Let $y_i = \frac{x_i}{1-p}$ and note that $\{y_i\}_{i>1}$ is a distribution. It follows that $\frac{A}{1-p}\frac{B}{1-p} \geq \alpha_2\beta_2$. In particular, $AB \geq (1-p)^2\alpha_1\beta_1$.

Next, let us handle the second summand in Equation (9). Let us re-write it:

$$\begin{aligned}\alpha_1 B + \beta_1 A &= (1-p)\sum_{i>1} \alpha_1 y_i \beta_i + \beta_1 y_i \alpha_i \\ &= (1-p)\,\alpha_1\beta_1 \sum_{i>1} y_i \left(\frac{\alpha_i}{\alpha_1} + \frac{\beta_i}{\beta_1}\right). \end{aligned} \tag{10}$$

We handle the term in big brackets using the assumption that $\alpha_1\beta_1 \leq \alpha_i\beta_i$. By this assumption it follows that $\frac{\alpha_i}{\alpha_1} \geq \frac{\beta_1}{\beta_i}$ and therefore $\frac{\alpha_i}{\alpha_1} + \frac{\beta_i}{\beta_1} \geq \frac{\beta_1}{\beta_i} + \frac{\beta_i}{\beta_1} \geq 2$. Plugging this into Equation (10), we obtain

$$\alpha_1 B + \beta_1 A \geq 2(1-p)\,\alpha_1\beta_1.$$

Using Equation (9) we obtain $f(\vec{x}) \geq p^2\,\alpha_1\beta_1 + 2p(p-1)\,\alpha_1\beta_1 + (1-p)^2\,\alpha_1\beta_1 = \alpha_1\beta_1$. ■

Appendix I. Crowd selection against the randomized benchmark

We design a crowd-selection algorithm with guarantees against the randomized benchmark. These guarantees are captured by Theorem 8 from Section 5, which we restate below for convenience. We focus on uniform costs, and (a version of) the single-crowd stopping rule from Section 2.

Our single-crowd stopping rule R_0 is as follows. Let $\widehat{\epsilon}_{*,t}$ be the empirical gap of the total crowd. Then R_0 stops upon reaching round t if and only if

$$\widehat{\epsilon}_{*,t} > C_{\texttt{qty}}/\sqrt{t} \quad \text{or} \quad t = T. \tag{11}$$

Here $C_{\texttt{qty}}$ is the "quality parameter" and T is a given time horizon.

Throughout this section, let $\mathcal{M}$ be the set of all distributions over crowds, and let $f^* = \max_{\mu\in\mathcal{M}} f(\mu)$ be the maximal induced gap. The benchmark cost is then at least $\Omega((f^*)^{-2})$.

We design an algorithm $\mathcal{A}$ such that $\texttt{cost}(\mathcal{A}|R_0)$ is upper-bounded by (essentially) a function of f^*, namely $O\left((f^*)^{-(k+2)}\right)$. We interpret this guarantee as follows: we match the benchmark cost for a distribution over crowds whose induced gap is $(f^*)^{2/(k+2)}$. By Lemma 6, the gap of the best crowd may be much smaller, so this is can be a significant improvement over the deterministic benchmark.

Theorem 16 (Theorem 8, restated) *Consider the bandit survey problem with uniform costs. Let R_0 be the single-crowd stopping rule given by (11). There exists a crowd-selection algorithm $\mathcal{A}$ such that $\texttt{cost}(\mathcal{A}|R_0) \leq O\left((f^*)^{-(k+2)}\sqrt{\log T}\right)$.*

In the rest of this section we prove Theorem 16. The proof relies on some properties of the induced gap: concavity and Lipschitz-continuity. Concavity is needed for the reduction lemma (Lemma 18), and Lipschitz-continuity is used to solve the MAB problem that we reduce to.

Claim 17 *Consider the induced gap $f(\mu)$ as a function on $\mathcal{M} \subset \mathbb{R}^k_+$. First, $f(\mu)$ is a concave function. Second, $|f(\mu) - f(\mu')| \leq n \left\|\mu - \mu'\right\|_1$ for any two distributions $\mu_1, \mu_2 \in \mathcal{M}$.*

Proof Let μ be a distribution over crowds. Then

$$f(\mu) = \mathcal{D}_\mu(x^*) - \max_{x \in \mathcal{O} \setminus \{x^*\}} \mathcal{D}_\mu(x) = \min_{x \in \mathcal{O} \setminus \{x^*\}} \mu \cdot \left(\vec{D}(x^*) - \vec{D}(x) \right). \tag{12}$$

Thus, $f(\mu)$ is concave as a minimum of concave functions. The second claim follows because $(\mu - \mu') \cdot \left(\vec{D}(x^*) - \vec{D}(x) \right) \leq n \left\|\mu - \mu\right\|_1$ for each option x. ■

Virtual rewards. Consider the MAB problem with virtual rewards, where arms correspond to distributions μ over crowds, and the virtual reward is equal to the induced gap $f(\mu)$; call it the *induced MAB problem*. The standard definition of regret is with respect to the best fixed arm, i.e. with respect to f^*. We interpret an algorithm $\mathcal{A}$ for the induced MAB problem as a crowd-selection algorithm: in each round t, the crowd is sampled independently at random from the distribution $\mu_t \in \mathcal{M}$ chosen by $\mathcal{A}$.

Lemma 18 *Consider the bandit survey problem with uniform costs. Let R_0 be the single-crowd stopping rule given by (11). Let $\mathcal{A}$ be an MAB algorithm for the induced MAB instance. Suppose $\mathcal{A}$ has regret $O(t^{1-\gamma} \log T)$ with probability at least $1 - \frac{1}{T}$, where $\gamma \in (0, \frac{1}{2}]$. Then*

$$\texttt{cost}(\mathcal{A}|R_0) \leq O\left((f^*)^{-1/\gamma} \sqrt{\log T} \right).$$

Proof Let $\mu_t \in \mathcal{M}$ be the distribution chosen by $\mathcal{A}$ is round t. Then the total crowd returns each option x with probability $\mu_t \cdot \vec{D}(x)$, and this event is conditionally independent of the previous rounds given μ_t.

Fix round t. Let $N_t(x)$ be the number times option x is returned up to time t by the total crowd, and let $\widehat{\mathcal{D}}_t(x) = \frac{1}{t} N_t(x)$ be the corresponding empirical frequency. Note that

$$\mathbb{E}\left[\widehat{\mathcal{D}}_t(x) \right] = \bar{\mu}_t \cdot \vec{D}(x), \quad \text{where } \bar{\mu}_t \triangleq \frac{1}{t} \sum_{s=0}^{t} \mu_s.$$

The time-averaged distribution over crowds $\bar{\mu}_t$ is a crucial object that we will focus on from here onwards. By Azuma-Hoeffding inequality, for each $C > 0$ and each option $x \in \mathcal{O}$ we have

$$\Pr\left[\left| \widehat{\mathcal{D}}_t(x) - \bar{\mu}_t \cdot \vec{D}(x) \right| < \frac{C}{\sqrt{t}} \right] > 1 - e^{-\Omega(C^2)}. \tag{13}$$

Let $\widehat{\epsilon}_t = \epsilon(\widehat{\mathcal{D}}_t)$ be the empirical gap of the total crowd. Taking the Union Bound in Equation (13) over all options $x \in \mathcal{O}$, we conclude that $\widehat{\epsilon}_t$ is close to the induced gap of $\bar{\mu}_t$:

$$\Pr\left[|\widehat{\epsilon}_t - f(\bar{\mu}_t)| < \frac{C}{\sqrt{t}} \right] > 1 - n\, e^{-\Omega(C^2)}, \quad \text{for each } C > 0.$$

In particular, R_0 stops at round t with probability at least $1 - \frac{1}{T}$ as long as

$$f(\bar{\mu}_t) > t^{-1/2} \left(C_{\texttt{qty}} + O(\sqrt{\log T})\right). \tag{14}$$

By concavity of f, we have $f(\bar{\mu}_t) \geq \bar{f}_t$, where $\bar{f}_t \triangleq \frac{1}{t}\sum_{s=0}^{t} f(\mu_s)$ is the time-averaged virtual reward. Now, $t\bar{f}_t$ is simply the total virtual reward by time t, which is close to f^* with high probability. Specifically, the regret of $\mathcal{A}$ by time t is $R(t) = t(f^* - \bar{f}_t)$, and we are given a high-probability upper bound on $R(t)$.

Putting this all together, $f(\bar{\mu}_t) \geq \bar{f}_t \geq f^* - R(t)/t$. An easy computation shows that $f(\bar{\mu}_t)$ becomes sufficiently large to trigger the stopping condition (14) for $t = O\left((f^*)^{-1/\gamma}\sqrt{\log T}\right)$. ■

Solving the induced MAB problem. We derive a (possibly inefficient) algorithm for the induced MAB instance. We treat $\mathcal{M}$ as a subset of $\mathbb{R}^k$, endowed with a metric $d(\mu, \mu') = n\,\|\mu - \mu'\|_1$. By Lemma 17, the induced gap $f(\mu)$ is Lipschitz-continuous with respect to this metric. Thus, in the induced MAB problem arms form a metric space $(\mathcal{M}, d)$ such that the (expected) rewards are Lipschitz-continuous for this metric space. MAB problems with this property are called *Lipschitz MAB* (Kleinberg et al., 2008).

We need an algorithm for Lipschitz MAB that works with virtual rewards. We use the following simple algorithm from (Kleinberg, 2004; Kleinberg et al., 2008). We treat $\mathcal{M}$ as a subset of $\mathbb{R}^k$, and apply this algorithm to $\mathbb{R}^k$. The algorithm runs in phases $j = 1, 2, 3, \ldots$ of duration 2^j. Each phase j is as follows. For some fixed parameter $\delta_j > 0$, discretize $\mathbb{R}^k$ uniformly with granularity δ_j. Let S_j be the resulting set of arms. Run bandit algorithm `UCB1` (Auer et al., 2002a) on the arms in S_j. (For each arm in $S_j \setminus \mathcal{M}$, assume that the reward is always 0.) This completes the specification of the algorithm.

Crucially, we can implement `UCB1` (and therefore the entire uniform algorithm) with virtual rewards, by using $\widehat{\epsilon}_t$ as an estimate for $f(\mu)$. Call the resulting crowd-selection algorithm `VirtUniform`.

Optimizing the δ_j using a simple argument from (Kleinberg, 2004), we obtain regret $O(t^{1-1/(k+2)}\,\log T)$ with probability at least $(1 - \frac{1}{T})$. Therefore, by Lemma 18 $\texttt{cost}(\texttt{VirtUniform}|R_0)$ suffices to prove Theorem 16.

We can also use a more sophisticated *zooming algorithm* from (Kleinberg et al., 2008), which obtains the same in the worst case, but achieves better regret for "nice" problem instances. This algorithm also can be implemented for virtual rewards (in a similar way). However, it is not clear how to translate the improved regret bound for the zooming algorithm into a better cost bound for the bandit survey problem.