POSITION PAPER

# Crowdsourcing Gold-HIT Creation at Scale: Challenges and Adaptive Exploration Approaches

Ittai Abraham Omar Alonso, Vasilis Kandylas, Rajesh Patel, Steven Shelford,
Aleksandrs Slivkins, Hai Wu[1]

October 2013

Technical Report
MSR-TR-2013-106

Gold HITs — Human Intelligence Tasks with known answers — are commonly used to measure worker performance and data quality in industrial applications of crowdsourcing. We suggest adaptive exploration as a promising approach for automated, scalable Gold HIT creation. We substantiate this with initial experiments in a stylized model.

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

[1]All: Microsoft. Contacts: {ittaia, omaralonso, vakandyl, rajeshpa, stevsh, slivkins, hawu}@microsoft.com.

Crowdsourcing has become a central tool for improving the quality of search engines and many other large-scale online services that require high quality data or labels. In this usage of crowdsourcing, a task or parts thereof are broadcast to multiple independent, relatively inexpensive workers, and their answers are aggregated. Automation and optimization of this process at a large scale allows to significantly reduce the costs associated with setting up, running, and analyzing experiments that contain such tasks.[1]

A typical crowdsourcing workload is partitioned into *HITs* (Human Intelligence Tasks), where each HIT has a specific, simple structure and involves only a small amount of work. This ensures consistency across multiple HITs and across multiple workers.

In many scenarios the task designer would benefit greatly by using a set of *Gold HITs* (a.k.a. *honey pots*). Gold HITs are a set of HITs where the associated answers are known in advance. They can be a very effective mechanism to measure the performance of workers and data quality. Previous research and our own experience have shown that task designers who use Gold HITs generally get high quality data on the experiments.

Gold HITs are usually generated manually, typically by hired domain experts. This approach is not scalable: it is expensive, time consuming and error prone. We believe that much more automated systems should be available, whereby a task designer starts with a relatively small Gold HIT set for bootstrapping, and uses the crowd to generate arbitrarily larger Gold HIT sets of high quality.

A central challenge in designing a system for automated Gold HIT creation is cost-efficient quality control. With error-prone workers, one needs to aggregate the answers of several workers to a obtain statistically robust answer for a Gold HIT. For cost-efficiency, one needs to take into account the heterogeneity in task difficulty and worker skills: some tasks are harder than others, and some workers are more skilled than others. Further, some workers may be more skilled at some tasks and less skilled at others, and the skill levels may vary over time (as a worker learns or becomes more/less motivated). In general, it is desirable to route tasks to the right skilled workers and use less aggregation for easier tasks and more skilled workers. The system initially has a very limited knowledge of worker skill and task difficulty, but they can be learned over time.

We measure each solution via the trade-off between *cost* and *quality*. In particular, we would like to obtain high quality (typically no more that 5% error relative to a panel of domain experts), while minimizing the costs, in terms of both time and money.

**Our approach: adaptive exploration.** Any good solution to the above challenge should involve *adaptive HIT assignment*, the assignment of HITs to workers which changes based on previous observations.[2] This raises a natural trade-off between *exploration* (experimentation to learn more about worker skill and task difficulty) and *exploitation*: making optimal decisions based on the experimentation results available so far. This trade-off occurs in many different scenarios, and is well-studied in Machine Learning and Operations Research (see [4, 3, 6] for background).

In many settings, the best algorithms for explore-exploit trade-off involve *adaptive exploration*, where not only the exploitation decisions, but the exploration schedule itself is adapted to the previous observations. For example, once we are sufficiently confident that a given alternative is bad, we can give up on it early and focus our exploration budget on more promising alternatives.

We believe that adaptive exploration is the right approach for automated Gold HIT creation. The technical challenge here is to incorporate the complexities of existing crowdsourcing scenarios, such as heterogeneity in workers/tasks and various practical and legal constraints.

[1] Large-scale crowdsourcing systems is an active research area. Surveying this research is beyond the scope of this position paper; see [7] for a forward-looking survey, and [8] for a more theoretical perspective.

[2] Adaptive HIT assignment can also be useful in settings other than Gold HIT creation.

**A concrete example.** To highlight the advantages of adaptive exploration, we consider a stylized model with heterogeneity in worker quality, but not HIT difficulty. The system processes a stream of HITs one by one. Each HIT is assigned to several workers, one by one, at unit cost per worker, until the Gold HIT answer is generated with sufficient confidence or the system gives up. The goal is to minimize the total cost while ensuring low error rate. For simplicity, let's assume that each HIT requires only binary answers.

We adopt the following idea from adaptive exploration: for each worker, combine exploration and exploitation in a single numerical score, updated over time, and at each decision point choose a worker with the highest current scores [9, 5, 2]. This score, traditionally called an *index*, takes into account both the average skill observed so far (to promote *exploitation*) and the uncertainty from insufficient sampling (to promote *exploration*). Over time, the algorithm zooms in on more skilled workers.

We use a simple algorithm which builds on the techniques from [2, 1]. For each worker $i$, let $t_i$ be the number of performed HITs for which the algorithm has generated a Gold HIT answer, and say in $t_i^+$ of those his answer coincides with the Gold HIT. If $t_i \geq 1$, we define this worker's index as

$$\texttt{IND}_i = \frac{t_i^+}{t_i} + \frac{1}{\sqrt{t_i}}.$$

(Note that $\texttt{IND}_i \leq 2$.) For initialization, we set $\texttt{IND}_i = 2$. At each time step, the algorithm picks a worker with the highest index, breaking ties arbitrarily.

For each HIT, we use a simple stopping rule from[1]: given $N_0$ "no" answers and $N_1$ "yes" answers, we stop and declare the majority answer to be the Gold HIT answer if $|N_0 - N_1| \leq C\sqrt{N_0 + N_1}$, i.e. if the majority answer is significant compared to random errors. Here $C$ is a pre-defined parameter.

**Experimental setup.** We use simulation parameterized by real data. We have $1,000$ workers. Each worker generates a correct answer for each HIT independently, with some fixed probability (*success rate*) which reflects his skill level. The success rate of each worker is drawn independently from a realistic "quality distribution" $\mathcal{D}_{\texttt{qty}}$.

We determined $\mathcal{D}_{\texttt{qty}}$ by examining a large set ($> 1,500$) of real workers on Microsoft's UHRS platform, and computing their average success rates over several months. Thus we obtained an empirical quality distribution, which we approximate by a low degree polynomial (see Figure 1).

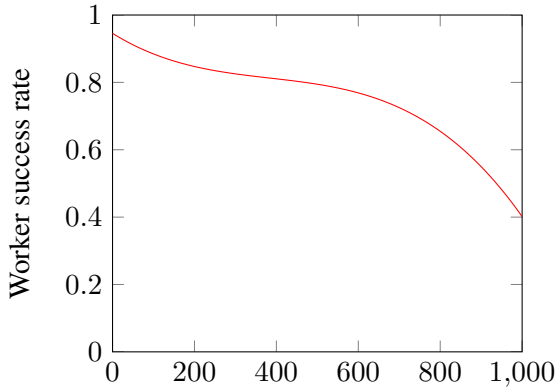


Figure 1: Worker quality distribution $\mathcal{D}_{\texttt{qty}}$.

We compare our index-based algorithm to a naive algorithm, called `Random`, which assigns each HIT to a random worker, using the same stopping rule as above. For both algorithms, we vary the parameter $C$

to obtain different cost vs. quality trade-offs. For each value of $C$, we compute $5K$ Gold HITs using each algorithm.

The simulation results are summarized in Figure 2. The main finding is that our index-based algorithm reduces the average cost by 35% to 50%. This suggest adaptive exploration, and particularly index-based algorithms, as a very promising approach for automated Gold HIT creation.

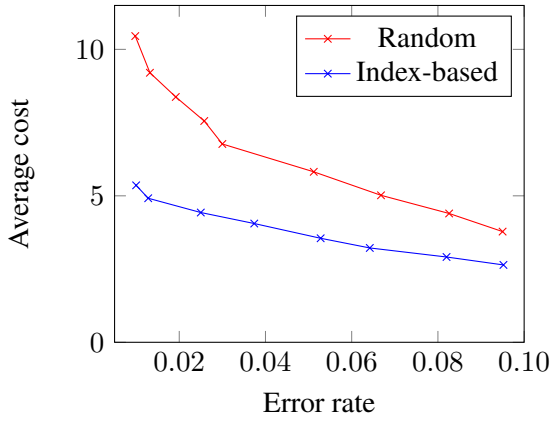


Figure 2: Simulation results.

# References


[1] Ittai Abraham, Omar Alonso, Vasilis Kandylas, and Aleksandrs Slivkins. Adaptive crowdsourcing algorithms for the bandit survey problem. In *26th Conf. on Learning Theory (COLT)*, 2013.

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

[3] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

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

[5] J. C. Gittins. Bandit processes and dynamic allocation indices (with discussion). *J. Roy. Statist. Soc. Ser. B*, 41:148–177, 1979.

[6] John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-Armed Bandit Allocation Indices*. John Wiley & Sons, 2011.

[7] Aniket Kittur, Jeffrey V. Nickerson, Michael S. Bernstein, Elizabeth M. Gerber, Aaron Shaw, John Zimmerman, Matthew Lease, and John J. Horton. The future of crowd work. In *16th ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW)*, 2013.

[8] Aleksandrs Slivkins and Jennifer Wortman Vaughan. Online decision making in crowdsourcing markets: Theoretical challenges, 2013. Position Paper. Available at `http://arxiv.org/abs/1308.1746`. To appear in *SIGEcom Exchanges*.

[9] 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.