# HiveMind: Tuning Crowd Response with a Single Value

***Preetjot Singh***[1], ***Walter S. Lasecki***[2], ***Paulo Barelli***[3], ***and Jeffrey P. Bigham***[4,2]

EECS Department
Northwestern University[1]
preet@u.northwestern.edu

Computer Science[2] / Economics[3]
University of Rochester
paulo.barelli@rochester.edu
wlasecki@cs.rochester.edu

Human Computer Interaction Institute[4]
Carnegie Mellon University
jbigham@cmu.edu

## Abstract

One common problem plaguing crowdsourcing tasks is tuning the set of worker responses: Depending on task requirements, requesters may want a large set of rich and varied worker responses (typically in subjective evaluation tasks) or a more convergent response-set (typically for more objective tasks such as fact-checking). This problem is especially salient in tasks that combine workers' responses to present a single output: Divergence in these settings could either add richness and complexity to the unified answer, or noise.

In this paper we present HiveMind, a system of methods that allow requesters to tune different levels of convergence in worker participation for different tasks simply by adjusting the value of one variable.

## Introduction

Crowdsourcing today has moved far beyond the traditional bounds of Mechanical Turk with applications such as VizWiz(Bigham et al. 2010) and Chorus(Lasecki et al. 2013c; 2013b). However, varied tasks bring varied goals: Having a conversation with a crowd may benefit from more varied responses, whereas picture captioning may require more convergent responses. However, task requirements can be nuanced: Perhaps while the desired goal of divergence is induced, the level of divergence may be deemed excessive by the requester.

In this paper we describe a system of payment algorithms that boxes a worker's actions in, consonant with the requester's direction. This system utilizes a carrot-and-stick approach: Workers financially gain if they perform the actions the requester pushes them towards, and lose if they perform any other action. This direction can be tuned by simply setting the value of one variable (called the $\alpha$-value). Crucially, this approach gets workers to optimize their efforts with respect to both time and accuracy. HiveMind also discourages *freeloaders* - workers who financially gain by playing actions arbitrarily, without any effort.

Optimizing a crowd with respect to both speed and accuracy is inherently difficult for three reasons: First, speed and accuracy are constraints on each other when it comes to optimization. Second, we have no way of knowing the abilities of individual workers or thus establish the optimal speed for responding or their capability for accuracy. Third, there may be no clear evaluation metric for crowd responses: These typically are associated with collective intelligence (CI) systems where the "best" or "correct" response is the one the majority (absolute or relative) of the crowd agrees with. CI tasks dominate crowdsourcing: Such tasks include image-labeling(von Ahn and Dabbish 2004), checking websites for qualitative content such as bias, hate speech or just situationally-inappropriate content(Attenberg, Ipeirotis, and Provost 2011; Aral, Ipeirotis, and Taylor 2011), and relevance of search results(Alonso, Kazai, and Mizzaro 2012).

HiveMind overcomes these problems by using the intuition that the best evaluator of a worker's capabilities, constraints and effort, is the worker herself[1].

## HiveMind

HiveMind is a system designed to tune convergence or divergence with a single value. It consists of two parts: (1) A stepwise structure for combining the crowd's responses. (2) A payment algorithm or method called *Pivot* that influences a worker to **Agree** (promoting convergence), **Propose** (promoting divergence) or abstain (reducing noise).

### The Setting

With HiveMind, in a task (such as labeling pictures), workers are presented with the input (the picture) and a set of choices (for labels).They can choose to vote for one these choices (i.e., play **Agree**), propose their own label (**Propose**), or simply abstain from responding.

Prior to the task, workers are selected by a bidding method (based on a Vickrey auction) whose main purpose is ensuring workers are invested in the outcome of the task by having a financial stake in it. This is the crux of the Pivot method: If workers see their payments differ with the effort they put in, they can be steered to put in effort on or away from specific actions depending on the task requirements.

### Stepwise Structure

Workers entering the task at any time are presented a set of choices (labels, in our example). Agreeing with a choice up-votes it and proposing a new choice gives other workers a chance to vote for it. The choice with the most votes is the output, and all workers who voted for it, and the one who proposed it, receive payoffs according to the algorithm. The stepwise structure has two main benefits: (1) It accommodates workers' variable entry and exit times, and (2), it

[1]We reasonably assume that workers have more information about their abilities and constraints than requesters do.

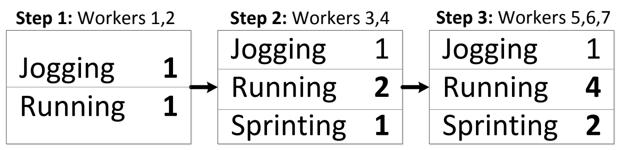

*Figure 1: Stepwise model used in an activity recognition domain: First, workers $w_1$ and $w_2$ create answers. Next, $w_3$ votes for 'Running', while $w_4$ creates another new answer. Finally, $w_5$ and $w_7$ vote for 'Running', $w_6$ votes for 'Sprinting'. At the end of the round, 'Running', created by $w_2$, wins.*

causes the input to evolve through this cascade of steps, presented a more refined input to every incoming set of workers.

## How Pivot Works

Workers are first selected by a bidding procedure based on the Vickrey auction. This ensures workers have a stake in the outcome of the task, and additionally, ensures the worst workers can do financially is within their comfort level.

Between the main actions of Agree and Propose, we can adjust the payoffs for each action in two ways: One, the potential reward for playing Propose relative to playing Agree is increased which promotes divergence of responses. Two, Propose's payoff relative to Agree's is decreased, promoting convergence. Either action is done by adjustment of a single value - $\alpha$.

Two main factors influence a worker's actions: (1) Her *confidence* - her opinion of her abilities, and (2) the payoff structure. In Pivot this worker confidence is defined with two variables: (i) $\mu = [0, 1]$ is the confidence level a worker has in playing agree and winning, and (ii) $\eta = [0, 1]$ is the confidence level a worker has in playing propose and winning.

In Pivot, the payoff method is structured as follows: Each selected worker stakes assigned amount $b$; The amount they can win by playing either Propose (P) or Agree (A) is a function of $b$: We adjust the relative payoffs by the value of $\alpha \in [0, 1]$. Then, $\text{payoff}_A = \mu b + (1-\mu)(-b)$
$\text{payoff}_P = \eta \frac{b}{\alpha} + (1-\eta)(-b)$
$\text{payoff}_{none} = 0$

## Causing a Worker to Play Agree or Propose

**Playing Agree** is feasible iff $\text{payoff}_A \geq 0$
$\implies \mu b + (1-\mu)(-b) \geq 0$
$\implies \mu \geq \frac{1}{2}$
Similarly, a worker will **play Propose** iff $\text{payoff}_P \geq 0$
$\implies \eta \frac{b}{\alpha} + (1-\eta)(-b) \geq 0$
$\implies \eta \geq \frac{\alpha}{1+\alpha}$
A worker **plays Propose over Agree** (when both are feasible) iff $\text{payoff}_P \geq \text{payoff}_A$
$\implies \eta \frac{b}{\alpha} + (1-\eta)(-b) \geq \mu b + (1-\mu)(-b)$
$\implies \eta \geq 2\mu\alpha$
If neither action is feasible, the worker **abstains** from acting.

## Freeloading and Worker Actions Over Time

A worker can play an action from her entry time $t_0$ until the end of the task. We stick with our assumption that the the worker (best) knows how much time (at least) to put in; let this time point be $t_1$. Under Pivot, playing without effort (prior to $t_1$) leads to a negative payoff (as the diagram

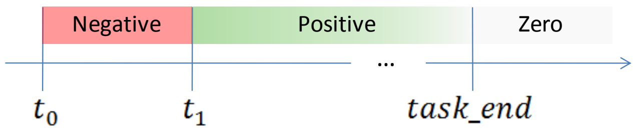

*Figure 2: From time $t_0$ to $t_1$ worker has negative expected payoff, the optimal point comes at time $t_1$, then declines until the task ends.*

shows), and playing after $t_1$ is a poor strategy since the task's exact end-time is not known.

## Applications and Future Work

HiveMind is most suited for crowd-powered applications with variable entry and exit times such as VizWiz(Bigham et al. 2010), Legion(Lasecki et al. 2011; 2013a), Scribe(Lasecki et al. 2012), and Adrenaline(Bernstein et al. 2011). Future and current work tests HiveMind's features with two sets of experiments: (1) Freeloading levels can be tested by inserting a blatantly incorrect choice and counting votes received, and (2) Measuring convergence by counting responses received with different $\alpha$-values. Game-theoretic mechanisms are often too complex for workers to understand, a common obstacle to their implementation. We propose to test this by comparing workers' responses to a given example situation with the theoretical best-response.

## Acknowledgements

This work was funded by the National Science Foundation and a Microsoft Research Ph.D. Fellowship.

## References

Alonso, O.; Kazai, G.; and Mizzaro, S. 2012. *Crowdsourcing for Search Engine Evaluation*. Springer.
Aral, S.; Ipeirotis, P.; and Taylor, S. 2011. Content and context: Identifying the impact of qualitative information on consumer choice. In Galletta, D. F., and Liang, T.-P., eds., *ICIS*. Association for Information Systems.
Attenberg, J.; Ipeirotis, P. G.; and Provost, F. J. 2011. Beat the machine: Challenging workers to find the unknown unknowns. In *HCOMP 2011*.
Bernstein, M. S.; Brandt, J.; Miller, R. C.; and Karger, D. R. 2011. Crowds in Two Seconds: Enabling Realtime Crowd-powered Interfaces. In *UIST 2011*, 33–42.
Bigham, J. P.; Jayant, C.; Ji, H.; Little, G.; Miller, A.; Miller, R. C.; Miller, R.; Tatarowicz, A.; White, B., W. S.; and Yeh, T. 2010. VizWiz: nearly real-time answers to visual questions. In *UIST 2010*, 333–342.
Lasecki, W. S.; Murray, K. I.; White, S.; Miller, R. C.; and Bigham, J. P. 2011. Real-time Crowd Control of Existing Interfaces. In *UIST 2011*, 23–32.
Lasecki, W. S.; Miller, C. D.; Sadilek, A.; AbuMoussa, A.; and Bigham, J. 2012. Real-time captioning by groups of non-experts. In *UIST 2012*.
Lasecki, W. S.; Song, Y.; Kautz, H.; and Bigham, J. P. 2013a. Real-Time Crowd Labeling for Deployable Activity Recognition. In *CSCW 2013*.
Lasecki, W. S.; Thiha, P.; Zhong, Y.; Brady, E.; and Bigham, J. P. 2013b. Answering Visual Questions with Conversational Crowd Assistants. In *ASSETS 2013*.
Lasecki, W.; Wesley, R.; Nichols, J.; Kulkarni, A.; Allen, J.; and Bigham, J. 2013c. Chorus: A crowd-powered conversational assistant. In *UIST 2013*. 2725–2730.
von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *CHI 2004*, 319–326.