Motivation-Aware Task Assignment in Crowdsourcing

Julien Pilourdault Sihem Amer-Yahia
Université Grenoble Alpes, CNRS, LIG, France
firstname.lastname@imag.fr

Dongwon Lee
Penn State University, USA
dongwon@psu.edu

Senjuti Basu Roy
NJIT, USA
senjutib@njit.edu

ABSTRACT

We investigate how to leverage the notion of *motivation* in assigning tasks to workers and improving the performance of a crowdsourcing system. In particular, we propose to model motivation as the balance between *task diversity*–i.e., the difference in skills among the tasks to complete, and *task payment*–i.e., the difference between how much a chosen task offers to pay and how much other available tasks pay. We propose to test different task assignment strategies: (1) RELEVANCE, a strategy that assigns *matching* tasks, i.e., those that fit a worker's profile, (2) DIVERSITY, a strategy that chooses *matching* and *diverse* tasks, and (3) DIV-PAY, a strategy that selects *matching* tasks that offer *the best compromise between diversity and payment.* For each strategy, we study multiple iterations *where tasks are re-assigned to workers as their motivation evolves.* At each iteration, RELEVANCE and DIVERSITY assign tasks to a worker from an available pool of filtered tasks. DIV-PAY, on the other hand, estimates each worker's motivation on-the-fly at each iteration, and uses it to assign tasks to the worker. Our empirical experiments study the impact of each strategy on overall performance. We examine both requester-centric and worker-centric performance dimensions and find that different strategies prevail for different dimensions. In particular, RELEVANCE offers the best task throughput while DIV-PAY achieves the best outcome quality.

1. INTRODUCTION

Crowdsourcing has become a popular framework to solve problems that are often hard for computers but easy for humans. Examples of crowdsourcing tasks include sentiment analysis in text, extracting information from images, and transcribing audio records. Despite recent successes, however, one of longstanding challenges in crowdsourcing is *task completion*–i.e., some tasks remain only partially completed or some workers do not work at full capacity. Studies have indicated that *workers' motivation* [25] plays a key role in task completion, especially since micro-tasks usually have very small monetary compensation. Therefore, it becomes increasingly important to *understand and model workers' motivation appropriately in the task assignment step.* In this paper, therefore, we study motivation in crowdsourcing and how to leverage it to assign tasks to workers. Our goal is to reach a better understanding of how to model motivation by studying its impact on different performance dimensions in crowdsourcing.

Existing literature has extensively studied how to perform task assignment to workers on crowdsourcing platforms [8, 17, 18, 23, 26]. Task assignment considers goals such as maximizing the quality of completed tasks, or minimizing task cost and latency to complete tasks. More recently, some research has reported noticeable improvement in task outcome quality when human factors, such as workers' skills and expected wage, were used in assigning tasks to workers [23, 26].

Yet, even when tasks are perfectly matched and assigned to workers initially, an important longstanding problem is *how to keep motivating workers who are not well-engaged in completing assigned tasks.* A recent ethnomethodological study on Turker Nation [22] argued that in order to maintain the attractiveness of crowdsourcing platforms, it is critical to enable *worker-centric optimization.* To address this problem, some existing work focused on incentivizing workers for long-lasting tasks [5, 19] or entertaining workers during task completion [7]. Moreover, recent studies have experimentally demonstrated the importance of intrinsic motivation in task completion [25]. While effective to some extent, these methods do not perceive task completion as an iterative process within which workers' motivation evolves, neither do they model that in the task assignment process. In this work, we advocate the need to *account for the evolution of workers' motivation* as workers complete tasks and capture that evolution in task assignment.

Our idea. Organization studies have explored worker motivation in physical workplaces since 70's [14]. Recently, some new efforts have examined and experimentally explored motivation on crowdsourcing platforms such as Amazon Mechanical Turk (AMT) [20, 25]. They have largely come to the conclusion that the motivation model developed in physical workplaces was also applicable in virtual marketplaces such as AMT.

Series ISSN: 2367-2005

10.5441/002/edbt.2017.23

Modeling workers' motivation is not obvious. While some workers may be driven by fun and enjoyment, others may look to advance their human capital, or increase their compensation. In fact, there are more than 13 factors that could be used to model motivation according to [20] (e.g., *task payment*, *task diversity*, *task autonomy*, *task identity*, *human capital advancement*, *pastime*). In addition, in a given session, a worker's motivation for a task may also depend on tasks that she has already completed and on other available tasks. In this work, as a first attempt to model workers' motivation and account for it in task assignment, we decided to focus on two factors: (1) *task diversity*, that is akin to skill variety, and (2) *task payment*. We believe that these two factors are good representatives of the spectrum of factors. Other factors will be examined in future.

Our formalization is grounded in the theory of work redesign [14]. The choice of diversity and payment allows us to clearly distinguish between "intrinsic" factors (e.g., how interested a worker is in the task's content) and "extrinsic" factors (e.g., how much the task pays). We can therefore use our formalization to verify which of intrinsic or extrinsic factors influence a worker's performance during task completion. Our formalization serves as a basis to define the **motivation-aware task assignment** (`Mata`) as a constrained optimization problem. Specifically, given a set of available tasks and a set of workers who are not working at full capacity, we identify which tasks are to be re-assigned to which worker, considering the worker's motivation. The worker-centric *and* adaptive nature of our problem make it novel. Indeed, while learning workers' skills in a crowdsourcing platform has been addressed before (e.g., [23]), leveraging those factors on-the-fly in a task assignment has not been addressed. There also exists a range of studies on online (iterative) task assignment but they consider only quality [8, 17, 18, 29] or budget and deadlines [11] in their objective. In fact, they rely on measuring the effectiveness of workers (e.g., in terms of *accuracy* [8, 29]) to adapt their assignment policy and do not consider the motivation of workers.

In contrast, our work considers motivation as a *first-class* factor in the objective function. Hence, none of the techniques proposed in the previous studies are applicable to our problem. To the best of our knowledge, our work is the first to propose to periodically revisit task assignment to workers by modeling and monitoring their motivation.

Our contributions. We propose to first formalize motivation factors that directly affect task completion. We then define our motivation-aware task assignment problem (`Mata`). We show that `Mata` is NP-hard using a reduction from the maximum dispersion problem (`MaxSumDisp`) [6, 10, 16, 24]. To solve our problem and isolate the effect of different dimensions on workers' motivation, we design and compare three task assignment strategies: (1) RELEVANCE, a strategy that chooses tasks that *match* a worker's profile, (2) DIVERSITY, a strategy that chooses *matching and diverse* tasks, and (3) DIV-PAY, a strategy that selects *matching* tasks with *the best compromise between diversity and payment.* DIV-PAY requires to observe workers as they complete tasks, estimate their motivation dynamically, and suggest the next most appropriate tasks. DIV-PAY is a $\frac{1}{2}$-approximation algorithm that uses a solution from the maximum diversification problem (`MaxSumDiv`), a general case of `MaxSumDisp`.

For each strategy, we study multiple iterations where tasks are re-assigned to workers. In order to compare our strategies, we develop a framework to hire workers from AMT and monitor them during task completion. In order to examine the effect of task diversity, we select $158,018$ micro-tasks released by CrowdFlower. Those tasks belong to 22 different kinds ranging from tweet classification to extracting information from news and assessing the sentiment of a piece of text. We measure common requester-centric dimensions such as task throughput (i.e., the number of tasks completed in multiple iterations per unit of time) and outcome quality with respect to a ground truth. We also measure dimensions that are considered both requester-centric and worker-centric, namely, worker retention (i.e., the number of workers who completed tasks) and payment. Worker motivation is measured as a worker-centric dimension.

Our empirical validation shows that different strategies prevail for different dimensions. RELEVANCE outperforms both DIV-PAY and DIVERSITY on task throughput and worker retention. However, DIV-PAY outperforms the other strategies on outcome quality. Workers completed more tasks and stayed longer when they were assigned tasks with RELEVANCE. That could be explained by the fact that very little context switching is required from workers in the case of RELEVANCE (since tasks are both relevant to the worker's profile and are potentially very similar to each other). DIVERSITY, on the other hand, is slightly inferior to DIV-PAY. That leads to the conclusion that diversity alone is not satisfactory as workers also pay attention to payment. The fact that DIV-PAY achieves the best outcome quality proves the need to actively monitor workers' motivation and incorporate it in task assignment. *Indeed, even if they are faster at completing similar tasks and stay longer in the system when tasks are relevant and not diverse, workers provide a higher-quality outcome for tasks that optimize their motivation, i.e., those chosen to achieve a balance between diversity and payment.* This confirms the need for worker-centric and adaptive approaches in crowdsourcing.

Paper organization. Section 2 formalizes the problem of motivation-aware crowdsourcing. Section 3 describes our three task assignment strategies. Section 4 reports performance results. Section 5 contains the related work. Finally, Section 6 summarizes our findings and their implications, and discusses possible future directions.

2. DATA MODEL AND PROBLEM

In this section, we first describe our model for tasks and motivation factors. Then, we formalize the motivation-aware task assignment problem. Table 1 summarizes important notations used throughout the paper.

2.1 Tasks and Workers

We consider a set of tasks $\mathcal{T} = \{t_1, \ldots, t_n\}$, a set of workers $\mathcal{W} = \{w_1, \ldots, w_p\}$ and a set of skill keywords $\mathcal{S} = \{s_1, \ldots, s_m\}$.

Notation	Definition
$\mathcal{T}$	a set of tasks $\{t_1, \ldots, t_n\}$
$\mathcal{W}$	a set of workers $\{w_1, \ldots, w_p\}$
$\mathcal{S}$	a set of skill keywords $\{s_1, \ldots, s_m\}$
$d(t_k, t_l)$	pairwise task diversity between two tasks
$\mathcal{T}_w^i$	tasks assigned to worker w at iteration i
$TD(\mathcal{T}')$	task diversity of a set of tasks $\mathcal{T}' \subseteq \mathcal{T}$
$TP(\mathcal{T}')$	task payment of a set of tasks $\mathcal{T}' \subseteq \mathcal{T}$
α_w	a worker w's relative importance between *task diversity* and *task payment*
$motiv_w(\mathcal{T}_w^i)$	the expected motivation of worker w on tasks $\mathcal{T}_w^i$
X_{max}	maximum number of tasks assigned to a worker
$matches(w, t)$	returns TRUE if the keywords of worker w *match* the keywords of task t

Table 1: A summary of important notations.

Tasks. A task t is represented by a vector $\langle t(s_1), t(s_2), \ldots, t(s_m), c_t\rangle$. For all $j \in [\![1, m]\!], t(s_j)$ is a Boolean value that denotes the presence or absence of skill keyword s_j in task t. The reward c_t is given to a worker who completes t. In our model, skill keywords may be interpreted as interests or qualifications, thereby allowing to capture a variety of tasks.

Workers. A worker w is represented by a vector $w = \langle w(s_1), \ldots, w(s_m)\rangle$. For all $j \in [\![1, m]\!], w(s_j)$ is a Boolean value capturing the interest of w in the skill keyword s_j.

EXAMPLE 1. Table 2 shows an example with 3 tasks, 2 workers and 5 skills. For instance, t_1 is characterized by a vector $\langle \texttt{true}, \texttt{true}, \texttt{false}, \texttt{false}, \texttt{false}, 0.01\rangle$: it is an audio transcription task with a \$0.01 reward, and it is described by skill keywords "*audio*" and "*English*". w_1 is a worker who expresses interest in tasks that feature the keywords "*audio*" and "*tagging*". We could suppose that only workers covering all task skills are qualified to complete a task. In this example, w_1 would only qualify for task t_2, while w_2 would qualify for both t_1 and t_3. □

2.2 Motivation Factors

Related work from the social sciences [20] on worker motivation in crowdsourcing includes 13 factors. The 6 most important ones are *Payment, Task Autonomy, Skill Variety, Task Identity, Human Capital Advancement, Pastime.* In this work, as a first attempt to model workers' motivation and account for it in task assignment, we focus on two factors: *task payment* and *task diversity*, that is akin to skill variety. Each factor is computed using a function that returns a motivation score. The choice of payment and diversity allows us to clearly distinguish between extrinsic motivation (payment) and intrinsic motivation (diversity) and offer enough variety in their values to study subtle differences in motivation. In addition, compared to other dimensions, only these two dimensions are most relevant in micro-tasks and in typical labor markets, such as Amazon Mechanical Turk. How to incorporate the remaining factors in modeling motivation is left to future work.

	audio	*English*	*French*	*review*	*tagging*	*reward* (\$)
t_1	✓	✓				0.01
t_2					✓	0.03
t_3			✓	✓		0.09
w_1	✓				✓	N/A
w_2	✓	✓	✓	✓		N/A

Table 2: Example of tasks and workers

Task Diversity. We denote the *pairwise task diversity* between two tasks t_k and t_l by $d(t_k, t_l)$. Pairwise task diversity essentially measures the aggregated differences of skills between two tasks. We ignore task reward in this definition. In our setting, we use the Jaccard similarity function $J()$ to define $d()$ as follows:

$$d(t_k, t_l) = 1 - J(\langle t_k(s_1), \ldots, t_k(s_m)\rangle, \langle t_l(s_1), \ldots, t_l(s_m)\rangle)$$

$d()$ is a metric and verifies the triangular inequality. We aim to be general and we do not fix one particular definition of $d()$ here. Instead, we allow any distance function (e.g., Euclidean distance, Jaro distance) as long as it verifies the triangular inequality. The *Task diversity* $TD(\mathcal{T}')$ of a set of tasks $\mathcal{T}' \subseteq \mathcal{T}$ is captured by aggregating the pairwise distances in $\mathcal{T}'$:

$$TD(\mathcal{T}') = \sum_{(t_k, t_l) \in \mathcal{T}'} d(t_k, t_l) \tag{1}$$

Task Payment. The total task payment of a set of tasks $\mathcal{T}' \subseteq \mathcal{T}$ is the sum of individual task payments in $\mathcal{T}'$:

$$TP(\mathcal{T}') = \frac{1}{\max_{t \in \mathcal{T}} c_t} \times \sum_{t \in \mathcal{T}'} c_t \tag{2}$$

The denominator $\max_{t \in \mathcal{T}} c_t$ normalizes each member of the sum in the interval $[0, 1]$.

2.3 Modeling a Worker's Motivation

We advocate a multi-step approach where the set of tasks assigned to a worker are revisited at each step in order to best fit the worker's motivation. At each iteration i, a worker w is assigned a new set of tasks $\mathcal{T}_w^i$. We wish to determine the best set $\mathcal{T}_w^i$ at each iteration i.

To capture the expected motivation of worker w on tasks in $\mathcal{T}_w^i$, we define a function $motiv_w^i$ as a linear[1] combination of diversity and payment of tasks in $\mathcal{T}_w^i$:

$$\begin{aligned} &motiv_w^i(\mathcal{T}_w^i) = \\ &2\alpha_w^i \times TD(\mathcal{T}_w^i) + (|\mathcal{T}_w^i| - 1)(1 - \alpha_w^i) \times TP(\mathcal{T}_w^i) \end{aligned} \tag{3}$$

[1] Note that we define the function as a linear formula between diversity and payment of a task, instead of a more complex non-linear formula, since a linear formula is likely to give rise to algorithms with theoretical guarantees as we show later, and is easier to interpret/explain.

α_w^i is a worker-specific parameter that reflects the relative importance between *task diversity* and *task payment*. We normalize the two components of the function with the factors $(|\mathcal{T}_w^i| - 1)$ and 2, since the first part of the sum counts $\frac{|\mathcal{T}_w^i|(|\mathcal{T}_w^i|-1)}{2}$ numbers and the second part $|\mathcal{T}_w^i|$ numbers [13]. We aim to accurately compute α_w^i that represents the compromise a worker w is looking for in choosing tasks to complete at each iteration i.

EXAMPLE 2. A worker w_1 with $\alpha_w^i = 0.1$ would be interested more in high-paying tasks with similar keywords (i.e., less diversity). This worker w_1 would choose tasks with a variety of keywords only if the payment is high enough. On the other hand, a worker w_2 with $\alpha_w^i = 0.9$ would be more motivated by task diversity. □

2.4 Problem

Now, we formally define the motivation-aware task assignment problem, `Mata`, as follows:

Problem 1 (Motivation-Aware Task Assignment) At each iteration i, and for each worker $w \in \mathcal{W}$, choose a subset of tasks $\mathcal{T}_w^i \subseteq \mathcal{T}$ such that:

$$\begin{aligned} &\textbf{max} && \mathit{motiv}_w^i(\mathcal{T}_w^i) & \\ &\textit{such that} && \forall t \in \mathcal{T}_w^i \ \mathit{matches}(w,t) & (C_1) \\ & && |\mathcal{T}_w^i| \leq X_{max} & (C_2) \end{aligned}$$ ■

The function $\mathit{matches}(w,t)$ in constraint C_1 returns `true` if the task t *matches* worker w. We can use various definitions for $\mathit{matches}(w,t)$. For instance, we can define $\mathit{matches}(w,t) =$ `true` iff the skill keywords of w and t are identical. In our setting, we suppose that $\mathit{matches}(w,t)$ captures how well the skill keywords of w *cover* the skill keywords of t. This allows us to capture cases where w *matches* t only if w expresses interest in at least 50% of the skill keywords of t. X_{max} is used in constraint C_2 to avoid assigning too many tasks to workers with varied interests, and reflects the ability of a worker to explore only a few tasks at a time (akin to limiting Web search results). Throughout this paper, we will suppose that each time we solve the `Mata` problem for a given worker w, w matches at least X_{max} tasks. Thus, given that the objective function is positive and monotonically increasing, w will be assigned *exactly* X_{max} tasks. That is a realistic assumption when X_{max} is chosen to be reasonably small (e.g., 20) in a context where we have a large collection of tasks to assign.

It is also important to note that the `Mata` problem considers each worker *independently*. When a worker w requires a new set of tasks $\mathcal{T}_w^i$, `Mata` is solved and tasks in $\mathcal{T}_w^i$ are dropped from $\mathcal{T}$. Thus, a task is assigned to at most one worker.

3. OUR APPROACHES

In order to study the effect of different dimensions in our problem, now, we explore approaches that exploit different objectives in the `Mata` problem. First, we design RELEVANCE, a diversity and payment-agnostic strategy. This strategy focuses on assigning to workers tasks that best match their interests. Second, we present DIV-PAY, that optimizes the objective function of the `Mata` problem. DIV-PAY is hence both diversity and payment-aware. Third, we present DIVERSITY, that focuses only on assigning diverse tasks to workers and is hence payment-agnostic.

Algorithm 1 RELEVANCE

```
Input: $\mathcal{T}$, $w, X_{max}, i$
Output: $\mathcal{T}_w^i$
$\mathcal{T}_w^i \leftarrow \emptyset$
$\mathcal{T}_{match(w)} \leftarrow \{t \in \mathcal{T} \mid matches(w,t)\}$
while $|\mathcal{T}_w^i| < X_{max}$
    $\mathcal{T}_w^i \leftarrow \mathcal{T}_w^i \cup \{\text{nextRandomTask}(\mathcal{T}_{match(w)} \setminus \mathcal{T}_w^i)\}$
return $\mathcal{T}_w^i$
```

3.1 Relevance strategy

We first propose the RELEVANCE approach (Algorithm 1), that assigns X_{max} random tasks that match workers' interests. RELEVANCE enforces constraints C_1 and C_2 and ignores task diversity and task payment. At each iteration i and for each worker w, RELEVANCE (i) filters the tasks $\mathcal{T}_{match(w)}$ that match w and (ii) selects randomly X_{max} tasks among $\mathcal{T}_{match(w)}$. In this strategy, a worker's motivation is therefore interpreted as matching her interests.

3.2 Diversity and payment-aware strategy

We present DIV-PAY, a strategy that is both diversity and payment-aware. DIV-PAY relies on both the on-the-fly estimation of a worker's motivation, and the online iterative task assignment. The motivation of a worker w is captured in the value of α_w^i, which represents the compromise a worker w is looking for in choosing tasks to complete at each iteration i. We first describe our approach to computing α_w^i. We then describe the task assignment algorithm that aims to solve the complete `Mata` problem.

3.2.1 Computing α_w^i

At each iteration i, we aim to learn α_w^i by leveraging tasks $t \in \mathcal{T}_w^{i-1}$ completed by worker w. Here, $\mathcal{T}_w^{i-1}$ refers to the tasks that were assigned to w in the previous iteration $(i-1)$ and that were presented to w as the set of available tasks. Let us consider the j-th task selected by worker w in $\mathcal{T}_w^{i-1}$. Each time when a worker selects a task t_j, we want to learn her preference for task diversity and task payment. To that purpose, we define α_w^{ij} to capture the compromise between skill variety and task payment made by the worker w when choosing t_j during iteration i. Our goal is to leverage a collection of such "micro-observations". First, we aim to capture each α_w^{ij}. Then, we aim to aggregate all α_w^{ij} to compute α_w^i.

We first show how to capture each α_w^{ij}. We start by considering each motivation factor independently. We suppose that when worker w chooses task t_j, she has already chosen tasks $\{t_1, \ldots, t_{j-1}\}$ where $j-1 \in [\![1, |\mathcal{T}_w^{i-1}|]\!]$.

Task Diversity. Equation 4 shows how we capture the gain in task diversity that a worker w seeks when picking a task

Algorithm 2 DIV-PAY

```
Input: $\mathcal{T}$, $w, X_{max}, i, \mathcal{T}_w^{i-1}, \{t_1, \dots, t_J\}$ tasks completed in iteration $i-1$
Output: $\mathcal{T}_w^i$
$\alpha_w^i \leftarrow \text{avg}_{k \in [\![1,J]\!]}\, \alpha_w^{ij}$
$\mathcal{T}_{match(w)} \leftarrow \{t \in \mathcal{T} \mid matches(w,t)\}$
$\mathcal{T}_w^i \leftarrow$ GREEDY($\mathcal{T}_{match(w)}, X_{max}, w$)
return $\mathcal{T}_w^i$
```

t_j in the remaining tasks $\mathcal{T}_w^{i-1} \setminus \{t_1, \dots, t_{j-1}\}$.

$$\Delta\, TD(t_j) = \frac{\sum\limits_{k \in 1,\dots,j-1} d(t_j, t_k)}{\max\limits_{t_{k'} \in \mathcal{T}_w^{i-1} \setminus \{t_1,\dots,t_{j-1}\}} \sum\limits_{k \in 1,\dots,j-1} d(t_{k'}, t_k)} \tag{4}$$

The numerator is the marginal gain in diversity when w selects a task t_j. The denominator reflects the maximum possible marginal gain when w selects a task in the remaining tasks $\mathcal{T}_w^{i-1} \setminus \{t_1, \dots, t_{j-1}\}$.

Task Payment. We compute the list of all *different* task payments in $\mathcal{T}_w^{i-1} \setminus \{t_1, \dots, t_{j-1}\}$ and sort it by descending order. Suppose that this list counts R elements and that $r(t_j)$ is the rank of c_{t_j} in this list (if c_{t_j} is the highest then $r(t_j) = 1$). We define *TP-Rank*$(t_j) \in [0,1]$ such that *TP-Rank*$(t_j) = 1$ *iff* t_j has the highest payment (0 if it has the lowest payment):

$$\textit{TP-Rank}(t_j) = 1 - \frac{r(c_{t_j}) - 1}{R - 1} \tag{5}$$

Equation 5 captures the willingness of w to choose tasks that pay highly among the available tasks.

EXAMPLE 3. Suppose that $\mathcal{T}_w^{i-1} \setminus \{t_1, \dots, t_{j-1}\} = \{t_5, t_6, t_7, t_8\}$ with $c_{t_5} = \$0.03, c_{t_6} = c_{t_7} = \$0.02, c_{t_8} = \$0.04$. A worker w selects t_5, which has the second highest task payment among the remaining tasks. We obtain *TP-Rank*$(t_5) = 1 - \frac{2-1}{3-1} = 0.5$. □

We have defined how to capture the importance of each factor. We now need to define α_w^{ij}, that captures the compromise between task diversity and task payment that worker w seeks when selecting task t_j. We set:

$$\alpha_w^{ij} = \frac{\Delta\, TD(t_j) + 1 - \textit{TP-Rank}(t_j)}{2} \tag{6}$$

α_w^{ij} is defined as the average of $\Delta\, TD(t_j)$ and $1 - \textit{TP-Rank}(t_j)$. The asymmetry comes from the fact that the higher α_w^i is, the lower is the importance of the task payment factor. We can observe that if both $\Delta\, TD(t_j)$ and $1 - \textit{TP-Rank}(t_j)$ return the same value, α_w^{ij} will be equal to 0.5.

Having defined each α_w^{ij}, we are now ready to capture α_w^i. Suppose that during iteration $i-1$ worker w chose J tasks where $J \leq |\mathcal{T}_w^{i-1}|$. We compute α_w^i as the average of all α_w^{ij}:

$$\alpha_w^i = \underset{k \in [\![1,J]\!]}{\text{avg}}\ \alpha_w^{ij} \tag{7}$$

Algorithm 3 GREEDY [4]

```
Input: $\mathcal{T}_{match(w)}, X_{max}, w, i$
Output: $\mathcal{T}_w^i$
$\mathcal{T}_w^i \leftarrow \emptyset$
while $|\mathcal{T}_w^i| < X_{max}$
    $t \leftarrow \arg\max_{t' \in \mathcal{T}_{match(w)} \setminus \mathcal{T}_w^i} g(\mathcal{T}_w^i, t')$
    $\mathcal{T}_w^i \leftarrow \mathcal{T}_w^i \cup \{t\}$
  return $\mathcal{T}_w^i$
```

3.2.2 Assigning Tasks

At each iteration i, the DIV-PAY strategy aims to solve the Mata problem for each worker. We first show that the Mata problem is NP-hard. Then, we present the DIV-PAY algorithm that returns a solution with an approximation ratio of 2 for the Mata problem.

Complexity. Intuitively, the Mata problem is difficult since its objective function includes the sum of pairwise distances, a common feature in several well-known NP-hard problems. In particular, Mata is closely related to the maximum dispersion problem (MaxSumDisp) [6, 10, 16, 24].

Theorem 1 *The motivation-aware task assignment problem (Mata) is NP-hard.* ■

PROOF. At each iteration and for each worker, the decision version of Mata is as follows. *Instance*: tasks $\mathcal{T}$, worker w and her α_w^i, X_{max} and an objective value Z. *Question*: is there a set $\mathcal{T}_w^i \subseteq \mathcal{T}$ such that C_1 is satisfied, $|\mathcal{T}_w^i| = X_{max}$ and $motiv_w^i(\mathcal{T}_w^i) \geq Z$? Mata $\in$ *NP* since a non-deterministic algorithm needs only to guess a set $\mathcal{T}_w$ and it can verify the question in polynomial time.

Reduction of Max Dispersion. To prove the NP-hardness, we consider the maximum sum dispersion problem (MaxSumDisp) [6, 10, 16, 24] (also known as *Maximum Edge Subgraph*)[2]. The decision version of this problem is as follows. *Instance*: a complete weighted graph $G = (V, E, \omega)$, an integer $k \in [2, |V|]$, an objective value Y. *Question*: Is there $V' \subseteq V$ such that $|V'| = k$ and $\sum_{v_1, v_2 \in V'} \omega(v_1, v_2) \geq Y$?

Note that MaxSumDisp is well-known to be NP-hard [4, 6, 10, 24] using a reduction from MaxClique [12]. Because a non-deterministic algorithm can guess a solution V' and easily verify it in polynomial time, MaxSumDisp $\in$ *NP*. Thus, MaxSumDisp is also NP-Complete. The reduction works as follows: each vertex $v \in V$ is mapped to a task $t_v \in \mathcal{T}$. The weight of an edge $(v_1, v_2) \in E$ is mapped to skill variety between two tasks: $\omega(\{v_1, v_2\}) = 2 * d(t_{v_1}, t_{v_2})$. We consider that $\alpha_w^i = 1$. We set $X_{max} = k$ and $Z = Y$. This creates an instance of Mata in polynomial time. This instance has the objective function $2 * \sum_{t_k, t_l \in \mathcal{T}_w^i} d(t_k, t_l)$. MaxSumDisp has a solution if and only if this instance of Mata has a solution. This proves the NP-hardness. □

Algorithm. Because it is an NP-hard problem, Mata is prohibitively expensive to solve on large instances. In our scenario, however, response time is important since Mata has to

[2] http://www.nada.kth.se/˜viggo/wwwcompendium/node46.html

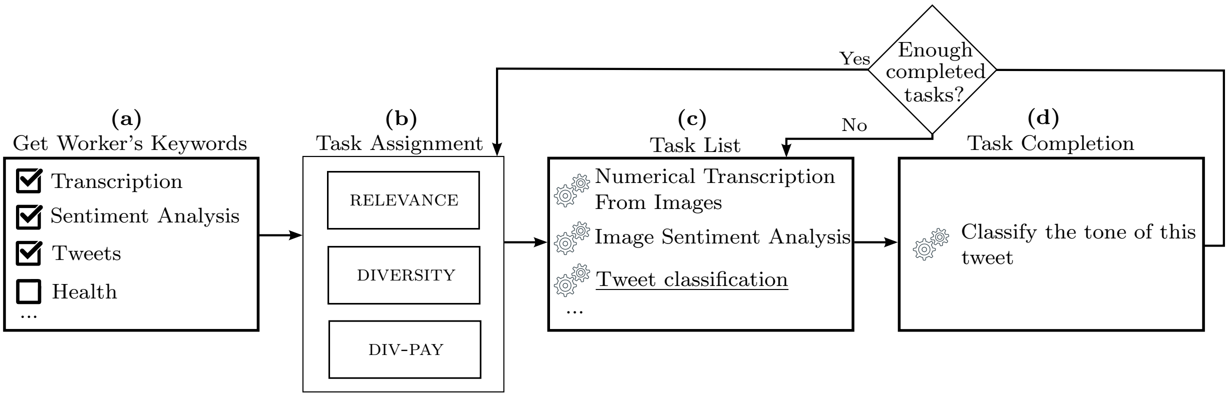


Figure 1: Workflow of our motivation-aware platform

be solved *online*, at each iteration i. The good news is that approximation algorithms exist for some related problems, such as `MaxSumDisp` [15, 16, 24] if the distance d satisfies the triangle inequality. The assumption that the distance obeys triangle inequality is not an overstretch, as many real world distances satisfy this assumption [3]. We consider that the pairwise task diversity is a metric and follows triangle inequality.

We adapt an existing algorithm for the maximum diversification problem `MaxSumDiv` (which includes `MaxSumDisp` as a special case). We design DIV-PAY (Algorithm 2) that assigns a set of tasks $\mathcal{T}_w^i$ to a worker w. First, DIV-PAY captures the motivation of the worker in the value of α_w^i. Then DIV-PAY computes a set of matching tasks (line 2) and runs GREEDY that returns $\mathcal{T}_w^i$ (line 3). GREEDY (Algorithm 3) is a $\frac{1}{2}$-approximation algorithm for the `MaxSumDiv` problem [4]. In the `MaxSumDiv` problem, the objective is to find a set S of p elements that maximizes

$$\lambda \sum_{(u,v)\in S} d(u,v) + f(S)$$

λ is a weight parameter, $f(S)$ is a normalized, monotone submodular function measuring the value of S and $d()$ is a distance function evaluating the diversity between two elements. Since `Mata` simplifies to finding a set of size *exactly* X_{max}, the objective function can be rewritten as:

$$motiv_w^i(\mathcal{T}_w^i) =$$
$$2\alpha_w^i \times TD(\mathcal{T}_w^i) + (X_{max}-1)(1-\alpha_w^i) \times TP(\mathcal{T}_w^i)$$

Now, we map our problem to the `MaxSumDiv` problem by setting $f(\mathcal{T}_w^i) = (X_{max}-1) \times (1-\alpha_w^i) \times TP(\mathcal{T}_w^i)$, $\lambda = 2\alpha_w^i$ and $p = X_{max}$. It can be easily seen that f is normalized ($f(\emptyset) = 0$). f is monotone since $\forall \mathcal{T}_1, \mathcal{T}_2 \subseteq \mathcal{T}$ s.t. $\mathcal{T}_1 \subseteq \mathcal{T}_2$ we have $f(\mathcal{T}_1) \leq f(\mathcal{T}_2)$. It is also submodular since $\forall \mathcal{T}_1, \mathcal{T}_2 \subseteq \mathcal{T}$ s.t. $\mathcal{T}_1 \subseteq \mathcal{T}_2$ and $\forall t \in \mathcal{T}$, we have:

$$f(\mathcal{T}_1 \cup \{t\}) - f(\mathcal{T}_1) = (X_{max}-1)(1-\alpha_w^i) \times \frac{1}{\max_{t'\in\mathcal{T}} c_{t'}} \times c_t$$
$$= f(\mathcal{T}_2 \cup \{t\}) - f(\mathcal{T}_2)$$

At each iteration, GREEDY inserts in $\mathcal{T}_w^i$ the task t that maximizes the function $g(\mathcal{T}_w^i, t) = \frac{1}{2}(f(\mathcal{T}_w^i \cup \{t\}) - f(\mathcal{T}_w^i)) + \lambda \sum_{t'\in\mathcal{T}_w^i} d(t,t')$ which is equal to $g(\mathcal{T}_w^i, t) = (X_{max}-1)(1-\alpha_w^i)TP(\{t\})/2 + 2\alpha_w^i \sum_{t'\in\mathcal{T}_w^i} d(t,t')$.

```
Algorithm 4 DIVERSITY
Input: $\mathcal{T}$, $w, X_{max}, i$
Output: $\mathcal{T}_w^i$
1: $\alpha_w^i \leftarrow 1$
2: $\mathcal{T}_{match(w)} \leftarrow \{t \in \mathcal{T} \mid matches(w,t)\}$
3: $\mathcal{T}_w^i \leftarrow$ GREEDY($\mathcal{T}_{match(w)}, X_{max}, w$)
4: return $\mathcal{T}_w^i$
```

We run GREEDY using tasks that verify the constraint C_1 (Algorithm 2, line 2), thus the algorithm returns a correct solution for the `Mata` problem. Because GREEDY is a $\frac{1}{2}$-approximation algorithm for the `MaxSumDiv` problem, DIV-PAY is a $\frac{1}{2}$-approximation for the `Mata` problem. Borodin et al. [4] observe that the GREEDY algorithm runs in time linear in the number of input elements when the desired size of the set is a constant. In our setting, we can conclude that DIV-PAY runs in $\mathcal{O}(X_{max} * |\mathcal{T}|)$ time.

One may wish to extend the motivation model used in `Mata`. We observe that the performance guarantee and the running time of GREEDY holds as long as our objective function has the form $\lambda \sum_{(u,v)\in S} d(u,v) + f(S)$ where f is a normalized, monotone and submodular function.

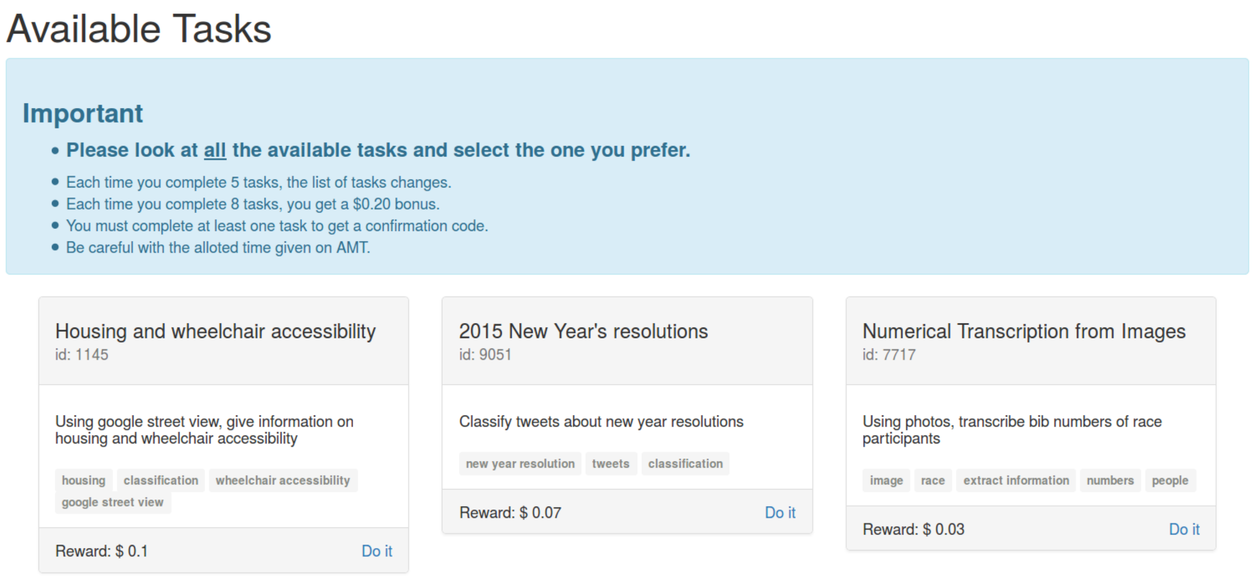
3.3 Diversity strategy

We propose DIVERSITY (Algorithm 4), a strategy that is diversity-aware and payment-agnostic. DIVERSITY considers a variant of the `Mata` problem where the objective includes only the task diversity sum. DIVERSITY employs GREEDY as a subroutine with $\alpha_w^i = 1$ at every iteration. We can follow the same reasoning exposed for `Mata`: constraint C_1 is enforced on line 2 and GREEDY is a $\frac{1}{2}$-approximation for the considered problem, so DIVERSITY is a $\frac{1}{2}$-approximation for this variant of the `Mata` problem.

4. EMPIRICAL VALIDATION

4.1 Workflow

We developed a web application to support motivation-aware crowdsourcing. Figure 1 illustrates a work session within our application. First, we get the interests of the worker w (Figure 1a). Then, we assign w a set of tasks (Figure 1b). Here, we can employ one of the three strate-



Available Tasks

Important

- **Please look at all the available tasks and select the one you prefer.**
- Each time you complete 5 tasks, the list of tasks changes.
- Each time you complete 8 tasks, you get a $0.20 bonus.
- You must complete at least one task to get a confirmation code.
- Be careful with the alloted time given on AMT.

Housing and wheelchair accessibility
id: 1145

Using google street view, give information on housing and wheelchair accessibility

housing classification wheelchair accessibility google street view

Reward: $ 0.1 Do it

2015 New Year's resolutions
id: 9051

Classify tweets about new year resolutions

new year resolution tweets classification

Reward: $ 0.07 Do it

Numerical Transcription from Images
id: 7717

Using photos, transcribe bib numbers of race participants

image race extract information numbers people

Reward: $ 0.03 Do it

Figure 2: Example screenshot of user interface – e.g., task grid

gies RELEVANCE, DIV-PAY or DIVERSITY. Then, the worker is presented a list of tasks (Figure 1c). She chooses a task and completes it (Figure 1d). If she has completed less than a pre-determined number of the X_{max} tasks, she is presented the same set of tasks again, except the tasks that were already completed. If she has completed enough tasks, another task assignment iteration is executed.The rationale behind imposing a minimum number of completed tasks before reiteration is to get a sufficient amount of input to accurately estimate α_w^i for each worker.

For the strategies RELEVANCE and DIVERSITY, we run the according algorithm at each iteration. For the strategy DIV-PAY, we need to consider the first iteration ($i = 1$) where w arrives for the first time on our platform. In this case, we cannot compute her α_w^1 since she has not yet completed tasks. In this first iteration, task assignment uses RELEVANCE as a cold-start assignment strategy. We aim to learn w's preference between diversity and payment using a strategy that does not favor any factor. Our rationale is to get an accurate estimation of α_w^1. On the next iterations, since w has already completed tasks, we run DIV-PAY. We compute her α_w^i and return the new set of tasks $\mathcal{T}_w^i$.

4.2 Settings

4.2.1 *Tasks*

We used a set of 158,018 micro-tasks released by Crowdflower [1]. It includes 22 different kinds of tasks, featuring for instance tweet classification, searching information on the web, transcription of images, sentiment analysis, entity resolution or extracting information from news. Each different kind of task is assigned a set of keywords that best describe its content and a reward, ranging from $0.01 to $0.12. Considered tasks are *micro-tasks* (they took on average 23s to be completed). We set payment proportional to the expected completion time.

4.2.2 *Task assignment*

We conducted experiments with all task assignment strategies, RELEVANCE, DIV-PAY, and DIVERSITY. We adapted the RELEVANCE strategy because the distribution of tasks is not uniform in our dataset (there are kinds of tasks that are over represented). The random task selection was achieved by first selecting a random *kind* of task, and then selecting a random task of this particular kind. We set $X_{max} = 20$ and imposed that 5 tasks must be completed before running another iteration. We set $matches(w, t) = \texttt{true}$ *iff* w is interested by at least 10% of the keywords of task t. Workers were asked to provide at least 6 keywords. We also verified the response time of our algorithms: any approach returned a solution in a few milliseconds upon a worker request. This makes our approaches suitable for an online setting: new workers and tasks can be easily handled by recomputing assignments from scratch.

4.2.3 *Workers and Payment*

We published 30 HITs on Amazon Mechanical Turk (AMT) to recruit workers. Each HIT corresponds to a work session on our platform. When a worker accept a HIT, she is asked to visit our web application. On our platform, she completes multiple tasks. When she terminates her work session, she get a verification code. Then, she paste the code on AMT and submit the HIT for payment. Each HIT may be submitted by at most 1 worker. Our HITs were completed by 23 different workers. We assigned 10 HITs for each task assignment strategy.

The HIT reward was set to $0.1. Each worker was granted a bonus equivalent to the total reward of the tasks she completed. To encourage workers who completed many tasks, we granted them a $0.2 bonus each time they completed 8 tasks. We required workers to have previously completed at least 200 HITs that were approved, and to have an approval rate above 80%. We also required HITs to be completed within 20 minutes: our rationale is to encourage workers to choose quickly tasks that they prefer.

4.2.4 User Interface

We conducted a first set of experiments where the $\mathcal{T}_w^i$ were displayed as a ranked list. We observed that most workers selected the top task first, completed it, and walked down the list in order. This created a bias and defeated our purpose: observing workers making choices based on their motivation. In order to reduce the effect of ranking, we changed the interface by showing a grid with 3 tasks per row (Figure 2). That mitigated the effect of ranking and workers stopped choosing the task in their order of appearance. We used the grid-based presentation in all our experiments.

4.2.5 Evaluation measures

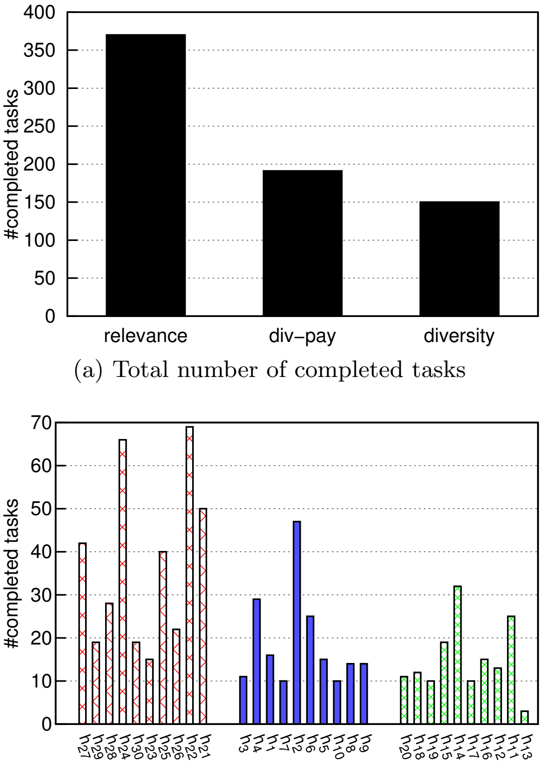
We evaluate our task assignment strategies using various measures. First, we consider *requester-centric* measures. Those include the *total number of completed tasks* across all iterations, and the *number of completed tasks per minute*, and *task throughput*, i.e., the number of completed tasks per work session. The higher the throughput, the faster a requester can have crowdwork completed. However, this measure does not reveal the quality of crowdwork. Hence, we also measure the *quality* of workers' contributions. Then, we consider measures that can be seen as both *requester* and *worker-centric*. That is the case for *task payment* and *worker retention* that quantifies the number of workers who completed tasks and captures the willingness of workers to work on our tasks. Finally, we measure *worker motivation*, a *worker-centric* dimension that quantifies workers' preferences between task diversity and task payment.

4.3 Results

23 different workers completed 711 tasks in 30 work sessions. On average, each worker spent 13 minutes to submit the HIT and completed 23.7 tasks. On average, 73% of workers chose fewer than 10 keywords (6 is the minimum possible).

4.3.1 Number of Completed tasks

Figure 3a presents the total number of completed tasks. Overall, RELEVANCE clearly outperforms DIV-PAY, which is slightly better than DIVERSITY. Figure 3b details the number of completed tasks for each work session h_k, $k \in [\![1, 30]\!]$. We observe that with RELEVANCE, 5 sessions had more than 40 completed tasks. With DIV-PAY and DIVERSITY, most workers completed fewer than 30 tasks. Figure 4 presents task throughput (i.e., number of completed tasks per min). We considered the total time spent on our application, including the time spent selecting a task to complete. The total time was higher with RELEVANCE (157 min) than with DIV-PAY (127 min). However, workers who were assigned tasks with RELEVANCE were more efficient (2.35 tasks/min vs 1.5 tasks/min). This could be explained by the fact that very little context switching is required from workers in the case of RELEVANCE (since tasks are both relevant to the worker and are potentially very similar to each other). DIVERSITY on the other hand, is slightly inferior to DIV-PAY. That leads us to the conclusion that workers did not necessarily appreciate diverse tasks, possibly for context-switching reasons. DIV-PAY slightly outperformed DIVERSITY, showing that including task payment as a motivating factor improves task throughput. While task throughput is a good indicator of the speed at which tasks are completed, it does not reveal the quality of crowdwork.



(a) Total number of completed tasks

(b) Number of completed tasks per work session

Figure 3: Number of completed tasks

4.3.2 Quality

For each kind of task, we sampled 50% of completed tasks and we manually evaluated their ground truth. We chose tasks for which defining a ground truth was not controversial (e.g., a task that asks for the presence or not of a pattern on an image). Then, we compared each worker's contribution to a task to its ground truth. Figure 5 presents the percentage of tasks that were correctly completed for each strategy. We observe that workers performed better with DIV-PAY (73% of correct answers) than with other strategies (DIVERSITY: 64%, RELEVANCE: 67%). This shows that assigning tasks that best match workers' compromise between task payment and task diversity encourages them to produce better answers. We observe that considering only task diversity (DIVERSITY) is not efficient. Including task payment is therefore important.

4.3.3 Worker retention

We now evaluate worker retention, a dimension that qualifies as both requester-centric and worker-centric. Figure 6a shows worker retention as the percentage of work sessions (vertical axis) that ended after x tasks were completed

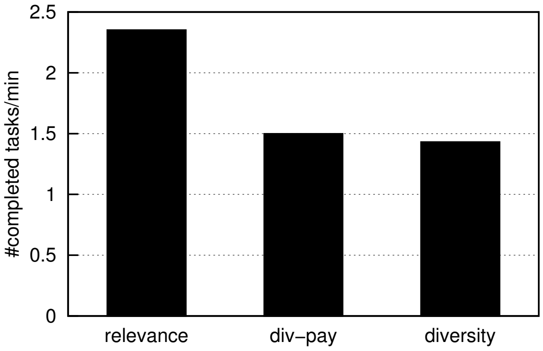


Figure 4: Task throughput

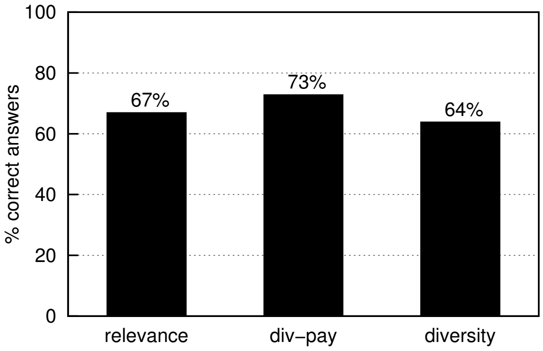
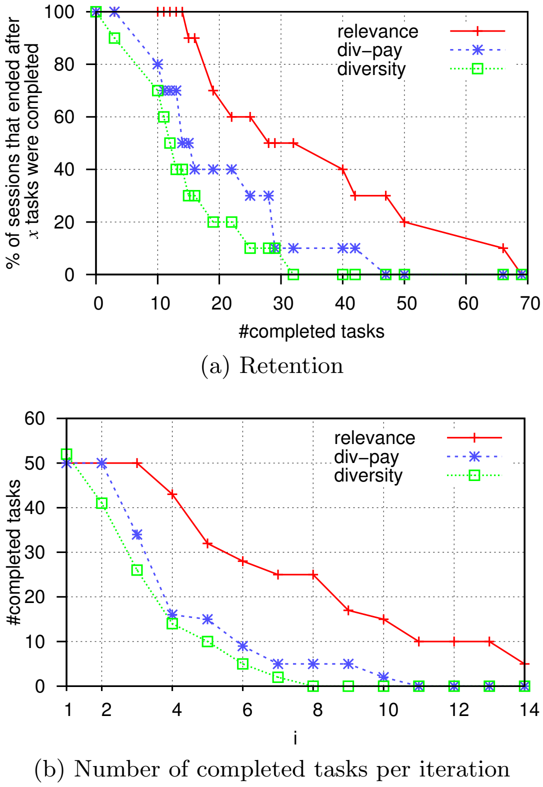


Figure 5: Evaluation of crowdwork quality

(horizontal axis). We find that workers stayed longer and completed more tasks when they were assigned tasks using RELEVANCE, hence this approach clearly outperforms DIV-PAY and DIVERSITY. Figure 6b supports this observation: more iterations were performed by workers with RELEVANCE. Although the number of completed tasks is roughly the same with all approaches on the first 2 iterations, this number fell quickly for DIV-PAY and DIVERSITY when $i > 2$. We also observe that DIV-PAY has a better worker retention than DIVERSITY. A plausible explanation is that workers are most comfortable completing similar tasks in a row. Therefore, they stay longer. They are least comfortable completing tasks with very different skills and tend to leave earlier. However, given that the quality of crowdwork reaches its best with DIV-PAY, we can say that optimizing for task relevance alone does not provide the best outcome quality even if workers are retained longer in the system.

4.3.4 Task Payment

Task payment is the other measure that qualifies as both *requester-centric* and *worker-centric*. Indeed, both requesters and workers look for a fair treatment when it comes to compensation. Requesters look to obtain high-quality contributions at a reasonable rate, and workers expect to be adequately paid for their efforts. Figure 7 presents the total task payment and the average payment per completed task for each strategy. The total payment (Figure 7a) is greater with RELEVANCE than with other approaches. This could be expected given the number of completed tasks (Section 4.3.1). However, the average task payment (Figure 7b) is the greatest with DIV-PAY. That is explained by the fact that DIV-PAY is the only strategy that is payment-aware. Thus, it is likely to assign higher-paying tasks to workers that prefer task payment over task diversity.



(a) Retention

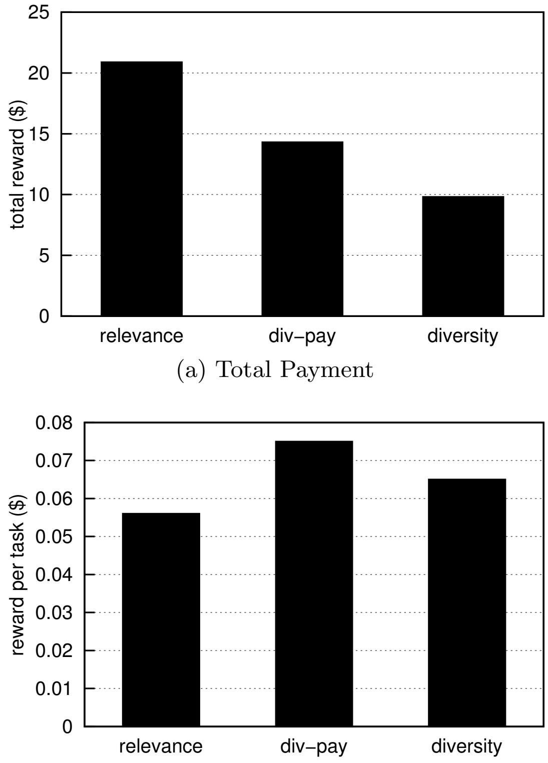
(b) Number of completed tasks per iteration

Figure 6: Worker retention and number of completed tasks per iteration.

4.3.5 Workers' motivation

We now turn to workers and study their motivation in detail. In order to make a fair comparison, we compute α_w^i for each strategy and for each iteration $i \geq 2$ (even if it is only used by DIV-PAY). Figure 8 shows the values of α_w^i for each work session h_k, $k \in 1 \ldots 30$. We omit session h_{13} (with DIVERSITY approach) where only 3 tasks were completed. We observe that in most work sessions, α_w^i oscillates around 0.5. Given the definition of α_w^i, this value indicates that most workers do not *steadily* favor task diversity over task payment. This is particularly observable on long work sessions in Figure 8a, where tasks were assigned using RELEVANCE. Figure 9 shows the distribution of α_w^i. It supports our observation: most workers do not make sharp choices. Most of the computed α_w^i values (72%) are in the interval $[0.3, 0.7]$, meaning that most workers do not favor task diversity over task payment, nor do they favor payment over diversity.

However, we do observe some sharp preferences for some workers. For instance, the worker in session h_2 (Figure 8b) clearly favored high-paying tasks. She completed 1.6 *different* tasks at each iteration (maximum possible: 5) that have an average reward of \$0.11 (maximum possible reward: \$0.12). Hence, her α_w^i was close to 0. Since she was assigned



(a) Total Payment

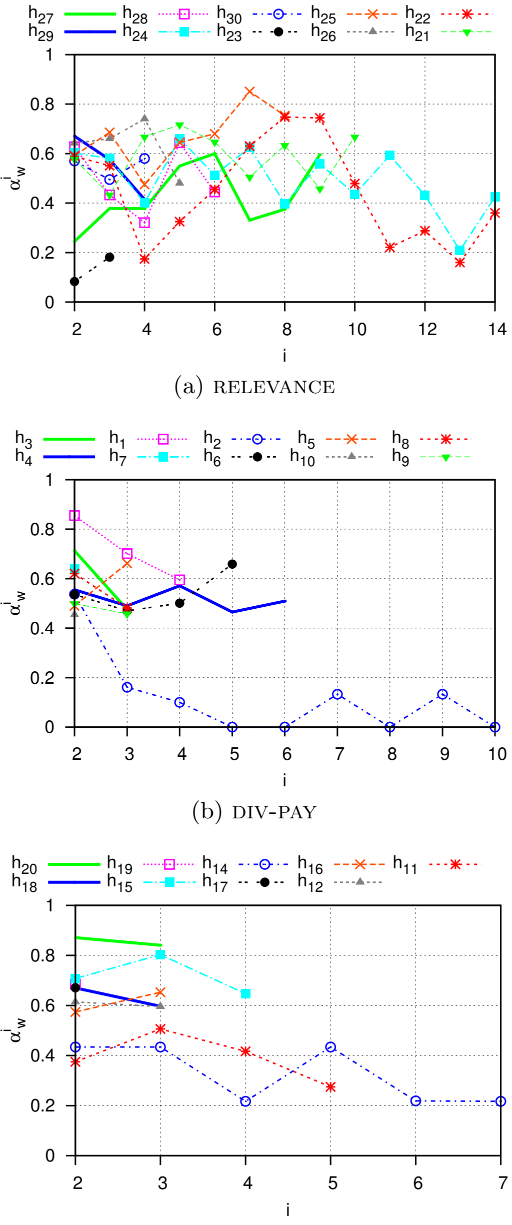
(b) Payment per task

Figure 7: Task payment

tasks using DIV-PAY, she received high-paying tasks. On the other hand, the worker in session h_{25} (Figure 8a) favored task diversity (her α_w^i is close to 0.8). She completed 4.12 *different* tasks at each iteration, that paid \$0.05 on average. This shows that our formulation allowed to accurately capture workers' preferences between task diversity and task payment.

4.4 Summary of Results and Discussion

We now summarize our results and provide a rationale for why different strategies prevail for different measures. Let us first consider *requester-centric* measures. We observe that RELEVANCE is the strategy that provides the best *task throughput*. This could be explained by the fact that RELEVANCE requires less effort from workers than DIVERSITY and DIV-PAY. Indeed, since RELEVANCE is based on selecting tasks that best match a worker's profile and since a worker's profile is quite homogeneous, tasks recommended by RELEVANCE are quite similar to each other. Therefore, a worker does not do much context switching between tasks and is hence faster overall. We also observe that DIV-PAY slightly outperforms DIVERSITY on task throughput. That shows the importance of task payment as an incentive. Results are different if we consider *crowdwork quality*. DIV-PAY is the strategy that obtains the best quality, followed by RELEVANCE. DIV-PAY is the only strategy that is both adaptive and motivation-aware: this contributes to providing a better incentive to workers. Quality comes at a price though: DIV-PAY is the strategy where the average *task payment* among completed tasks is the highest and it does not provide the



(a) RELEVANCE

(b) DIV-PAY

(c) DIVERSITY

Figure 8: Evolution of α_w^i (h_k is k-th work session)

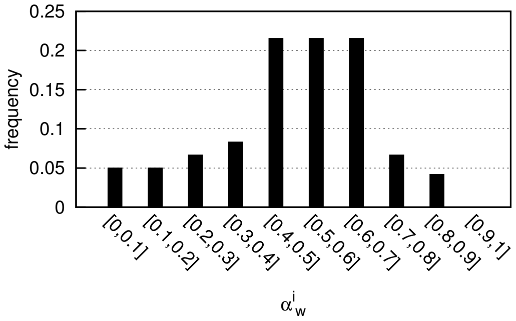


Figure 9: Distribution of α_w^i

highest task throughput (it is better than DIVERSITY but worse than RELEVANCE). Thus, depending on the platform, one should study trade-offs between these strategies when designing task assignment algorithms.

If we consider *worker-centric* measures, we observe that DIV-PAY is the best strategy for *task payment* since it rewarded workers higher. This could be expected since it is the only payment-aware strategy. However, *worker retention* is better with RELEVANCE. Workers performed longer work sessions with RELEVANCE than with other strategies. This finding is related to the fact that most workers do not have a clear preference for task diversity or task payment. They prefer tasks that match their interests and require fewer context switching, hence they did not necessarily stay longer when task diversity or task payment were favored (with DIV-PAY or DIVERSITY). We also observe *workers' motivation* and we notice that some workers carefully choose task diversity or task payment. In that case, we could accurately capture their preferences with appropriate α_w^i values. That allowed DIV-PAY to slightly outperform DIVERSITY on both task throughput and worker retention.

5. RELATED WORK

Task Assignment in Crowdsourcing. Task assignment in crowdsourcing was largely studied. Previous studies notably include the design of *adaptive* algorithms, that focus on maximizing the quality of crowdwork [8, 17, 18, 29]. For instance, Fan et al. [8] leverage the similarity between tasks and the past answers of workers to design an adaptive algorithm that aims at maximizing the accuracy of crowdwork. Ho et al. [17] study an online setting, where the workers who are going to arrive on the platform have unknown skill. They design an algorithm where the skill level of sampled workers is observed and is leveraged to assign all other workers to tasks. None of these studies includes motivation factors in their model. Other investigations focused on dynamically adjusting the task reward [9, 11] so as to satisfy a deadline or a budget constraint. They modeled the willingness of a worker to choose a task as a task acceptance probability featuring task reward as a parameter. These studies do not focus on task assignment as workers self-appoint themselves to tasks and they do not include task diversity in their model. Recently, Rahman et al. [23] focused on assigning tasks to groups of *diverse workers* in collaborative crowdsourcing. Moreover, Wu et al. aim at finding sets of workers with diverse *opinions* [28]. None of those studies assigns *diverse tasks* to workers or includes motivation factors. Moreover, Rahman et al. [23] consider a setting which is not *adaptive*: task assignment does not leverage previous answers to improve the main objective.

Motivating Workers. A range of studies point out the importance of suitably motivating workers in crowdsourcing [2, 21, 22]. Obviously, reward is an important factor, and crowdsourcing platform should follow some guidelines that would solve wage issues [2]. Kittur et al. [21] underlines the interest of designing frameworks that include incentive schemes other than financial ones. In particular, they notice that a system should "*achieve both effective task completion and worker satisfaction*". Worker motivation was first studied in physical workplaces [14]. Recent studies [20] investigated the importance of 13 motivation factors for workers on Amazon Mechanical Turk. Although task payment remains the most important factor, Kaufmann et al. [20] point out that workers are also interested in *skill variety* or *task autonomy.*

Some efforts were driven towards experimenting motivation factors in crowdsourcing [5, 7, 25, 27]. In a recent study [7], Dai et al. inserted *diversions* in the workflow such that workers were presented with some entertainment contents between two task completions. Dai et al. showed that such a motivational scheme improved worker retention. Chandler and Kapelner [5] conducted experiments showing that workers perceiving the "meaningfulness" of task improved throughput without degrading quality. Another study [25] assessed the effect of extrinsic and intrinsic motivation factors. They demonstrated that workers were more accurate on meaningful tasks posted by a non-profit organization than on tasks posted by a private firm and less explicit about their outcome. This suggested that intrinsic factors help improve quality of crowdwork. Shaw et al. [27] assessed 14 incentives schemes and found that incentives based on worker-to-worker comparisons yield better crowdwork quality. None of the above studies leverages motivation factors to optimize task assignment, and thus they do not tackle the motivation-aware task assignment problem.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented three different formulations of task assignment algorithms and compared three motivation-aware task assignment strategies: RELEVANCE, DIVERSITY, and DIV-PAY. Both RELEVANCE and DIVERSITY are based on matching tasks to a worker's interests, while DIV-PAY relies on assigning tasks to workers based on their observed motivation. This work builds on the premise that workers may look for a balance between intrinsic motivation, modeled as task diversity, and extrinsic motivation, modeled as task payment.

In practice, our experiments show that different strategies prevail. RELEVANCE offers the best task throughput and worker retention since it requires less context switching for workers (i.e., all tasks are similar). DIV-PAY, however, has the best outcome quality, since it allows workers to achieve the best compromise between fun and compensation. This last observation is important as it confirms that even when workers are slower at executing tasks and when they spend less time on a platform, optimizing for their motivation impacts task outcome quality positively. There are also a few cases where DIV-PAY outperforms DIVERSITY on task throughput and worker retention, implying the need to account for payment in addition to diversity. Those are the cases of workers whose preference between diversity and payment are sharply expressed as they choose tasks to complete. Our results highlight the importance to understand the evolving motivation of workers on a crowdsourcing platform and the importance to measure both requester-centric and worker-centric dimensions.

In the future, we would like to investigate the possibility of making the platform transparent by showing to workers what the system learned about them and letting them provide explicit feedback. We will also investigate the impact of other motivation factors such as "social signaling" and "advancing human capital" with respect to measures such as task throughput and worker retention.

References

[1] Crowdflower - data for everyone. https://www.crowdflower.com/data-for-everyone/.

[2] B. B. Bederson and A. J. Quinn. Web workers unite! addressing challenges of online laborers. In *CHI*, pages 97–106, 2011.

[3] A. Besicovitch. On a general metric property of summable functions. *Journal of the London Mathematical Society*, 1(2):120–128, 1926.

[4] A. Borodin, H. C. Lee, and Y. Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *PODS*, pages 155–166, 2012.

[5] D. Chandler and A. Kapelner. Breaking monotony with meaning: Motivation in crowdsourcing markets. *CoRR*, abs/1210.0962, 2012.

[6] B. Chandra and M. M. Halldórsson. Approximation algorithms for dispersion problems. *J. Algorithms*, 38(2):438–465, 2001.

[7] P. Dai, J. M. Rzeszotarski, P. Paritosh, and E. H. Chi. And now for something completely different: Improving crowdsourcing workflows with micro-diversions. In *ACM CSCW*, pages 628–638, 2015.

[8] J. Fan, G. Li, B. C. Ooi, K.-l. Tan, and J. Feng. icrowd: An adaptive crowdsourcing framework. In *SIGMOD*, pages 1015–1030, 2015.

[9] S. Faradani, B. Hartmann, and P. G. Ipeirotis. What's the right price? pricing tasks for finishing on time. In *AAAI*, 2011.

[10] S. P. Fekete and H. Meijer. Maximum dispersion and geometric maximum weight cliques. *CoRR*, cs.DS/0310037, 2003.

[11] Y. Gao and A. G. Parameswaran. Finish them!: Pricing algorithms for human computation. *PVLDB*, 7(14):1965–1976, 2014.

[12] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[13] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW*, pages 381–390, 2009.

[14] J. Hackman and G. R. Oldham. Motivation through the design of work: Test of a theory. *Organizational Behavior and Human Performance*, 16(22):250–279, 1976.

[15] R. Hassin and S. Rubinstein. An improved approximation algorithm for the metric maximum clustering problem with given cluster sizes. *Inf. Process. Lett.*, 98(3):92–95, 2006.

[16] R. Hassin, S. Rubinstein, and A. Tamir. Approximation algorithms for maximum dispersion. *Oper. Res. Lett.*, 21(3):133–137, 1997.

[17] C. Ho, S. Jabbari, and J. W. Vaughan. Adaptive task assignment for crowdsourced classification. In *ICML*, pages 534–542, 2013.

[18] C. Ho and J. W. Vaughan. Online task assignment in crowdsourcing markets. In *AAAI*, 2012.

[19] J. J. Horton and L. B. Chilton. The labor economics of paid crowdsourcing. In *ACM EC*, pages 209–218, 2010.

[20] N. Kaufmann, T. Schulze, and D. Veit. More than fun and money. worker motivation in crowdsourcing - A study on mechanical turk. In *AMCIS*, 2011.

[21] A. Kittur, J. V. Nickerson, M. S. Bernstein, E. Gerber, A. D. Shaw, J. Zimmerman, M. Lease, and J. Horton. The future of crowd work. In *CSCW*, pages 1301–1318, 2013.

[22] D. B. Martin, B. V. Hanrahan, J. O'Neill, and N. Gupta. Being a turker. In *CSCW*, pages 224–235, 2014.

[23] H. Rahman, S. B. Roy, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Task assignment optimization in collaborative crowdsourcing. In *IEEE ICDM*, pages 949–954, 2015.

[24] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42(2):299–310, 1994.

[25] J. Rogstadius, V. Kostakos, A. Kittur, B. Smus, J. Laredo, and M. Vukovic. An assessment of intrinsic and extrinsic motivation on task performance in crowdsourcing markets. In *ICWSM*, 2011.

[26] S. B. Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Task assignment optimization in knowledge-intensive crowdsourcing. *VLDB J.*, 24(4):467–491, 2015.

[27] A. D. Shaw, J. J. Horton, and D. L. Chen. Designing incentives for inexpert human raters. In *CSCW*, pages 275–284, 2011.

[28] T. Wu, L. Chen, P. Hui, C. J. Zhang, and W. Li. Hear the whole story: Towards the diversity of opinion in crowdsourcing markets. *PVLDB*, 8(5):485–496, 2015.

[29] Y. Zheng, J. Wang, G. Li, R. Cheng, and J. Feng. QASCA: A quality-aware task assignment system for crowdsourcing applications. In *SIGMOD*, pages 1031–1046, 2015.