Proceedings, The Fourth AAAI Conference on
Human Computation and Crowdsourcing

Extending Workers' Attention Span Through Dummy Events

Avshalom Elmalech
Harvard University, USA
elmalech@seas.harvard.edu

David Sarne
Bar-Ilan University, Israel
sarned@cs.biu.ac.il

Esther David
Ashkelon Academic College, Israel
astrdod@acad.ash-college.ac.il

Chen Hajaj
Vanderbilt University, USA
chen.hajaj@vanderbilt.edu

Abstract

This paper studies a new paradigm for improving the attention span of workers in tasks that heavily rely on user's attention to the occurrence of rare events. Such tasks are highly common, ranging from crime monitoring to controlling autonomous complex machines, and many of them are ideal for crowdsourcing. The underlying idea in our approach is to dynamically augment the task with some dummy (artificial) events at different times throughout the task, rewarding the worker upon identifying and reporting them. This, as an alternative to the traditional approach of exclusively relying on rewarding the worker for successfully identifying the event of interest itself. We propose three methods for timing the dummy events throughout the task. Two of these methods are static and determine the timing of the dummy events at random or uniformly throughout the task. The third method is dynamic and uses the identification (or misidentification) of dummy events as a signal for the worker's attention to the task, adjusting the rate of dummy events generation accordingly. We use extensive experimentation to compare the methods with the traditional approach of inducing attention through rewarding the identification of the event of interest and within the three. The analysis of the results indicates that with the use of dummy events a substantially more favorable tradeoff between the detection (of the event of interest) probability and the expected expense can be achieved, and that among the three proposed method the one that decides on dummy events on the fly is (by far) the best.

Introduction

In many monitoring tasks, workers are requested to respond and take action only upon identifying some rare or unusual events that occur very infrequently (if at all) along the task. Examples for such tasks include tracking satellite broadcasted data with the aim of detecting events of satellite malfunction, watching suitcases passing through an X-ray machine (e.g., at airports) with the aim of detecting sharp objects or explosives, network monitoring at a Network Operations Center (NOC) with the aim of identifying service degradation or failure, and watching streams arriving from Closed Circuit Televisions (CCTVs) with the aim of identifying crime. The performance of a worker in such tasks is critically correlated with the extent she is tuned to the continuous sequence of events, as even with the slightest loss of attention the event of interest may be missed. Furthermore, common to all the above examples, that the work is highly monotonous and normally workers' attention degrade with time (Rahman 2012).

Many of these tasks can potentially be or are already in the process of being outsourced using crowdsourcing. This is mostly due to the ability to recruit fast-reacting paid crowds within seconds upon demand (Bernstein et al. 2012; Yin, Chen, and Sun 2014; Yin and Chen 2015; Doan, Ramakrishnan, and Halevy 2011; Bernstein et al. 2011). One prominent example for this trend is CCTV surveillance. The emerging globalized phenomenon of huge urban areas covered by CCTV surveillance, which initially was used mainly in the private sector, now appears to be rapidly diffusing into the public realm resulting in a massive amount of unwatched or unprocessed videos (Norris, McCahill, and Wood 2004).[1] In that sense, citizens monitoring of CCTV feeds is a novel development. Trottier (2014) provides an exploratory overview of crowdsourced surveillance of closed-circuit television (CCTV) footage over the Internet. He considers four case studies of UK-based crowdsourced CCTV surveillance on the Internet, among which the most relevant to our case is *Internet Eyes* which acts as an intermediary between businesses that need personnel to sort through their CCTV footage, and individuals who want to work in this area.

The common and most straightforward method for keeping workers tuned to their monitoring task, hence increasing the chance of identifying events of interests is rewarding workers upon successful identification of such events (Ho et al. 2015; Yin and Chen 2015; Yin, Chen, and Sun 2014). The proposed reward creates an incentive to keep focused in the task, regardless of how monotonous it becomes, as missing the event of interest, e.g., because of switching to another application while keeping the crowdsourced task in the background, is now associated with some potential loss. The greater the proposed reward the greater the potential loss (or "regret" (Zeelenberg and Beattie 1997; Loomes and Sugden 1982; Levy and Sarne 2016)) associated with missing the event of interest and consequently the greater the incentive to keep focused throughout the task.

[1] The UK is arguably the most surveilled nation via CCTV-with approximately 1.85 million private and public cameras (Trottier 2014).

In this paper we propose an alternative approach for overcoming the degradation in workers' attention span over time in monitoring tasks. Our approach is based on intelligent insertion of artificial ("dummy") events, rewarding the worker if successfully identifying them. Namely, the worker is told that other than the main event of interest, there are some additional well-defined events that she should identify and report as these encapsulate additional monetary rewards. While the underlying idea itself is quite simple the challenging aspect of the proposed method is the determination of when to introduce dummy events. A frequent introduction of dummy events will surely keep the worker in focus (assuming a reasonable reward per dummy event identification), however at the same time will substantial increase the expected expense over the task overall. We propose and provide a thorough evaluation of three methods for generating dummy events. The first two suggest a simple scheme of introducing a prespecified number of dummy events either at random times or uniformly (i.e., at fixed intervals) throughout the task. The advantage of these schemes is mainly in the bound they put on the payment to the worker, as the number of dummy events generated is fixed and pre-determined. The third method we propose is inherently dynamic. It uses the dummy events (and their identification and misidentification) for modeling the worker's attentional state and makes decisions concerning the introduction of additional dummy events based on this measure, on the fly. This way, dummy events are introduced only when necessary, resulting in lower expected expense overall.

To test our methods we used an online game designed in a way that the worker's attention to what is displayed on the screen directly reflects on her success to identify the event of interest. Workers, recruited through Amazon Mechanical Turk (AMT), received a small fixed payment for participating in the experiment, a bonus for identifying the principle event of interest and a smaller bonus for each dummy event they identified. The results were compared to those obtained when not using dummy events at all and within the three proposed methods. Overall, all three methods where found to perform better than when not using dummy events at all, in the sense that they provide a better tradeoff between the expected payment for the task and the chance of identifying the primary event of interest. Among the three methods, the one that generates dummy events dynamically was found to be significantly better than the other two.

Related Work

In the past decade we have witnessed a plethora of crowdsourcing systems (also termed collective intelligence, wikinomics, crowd wisdom, human computation and more (Doan, Ramakrishnan, and Halevy 2011)) from various application domains (Franklin et al. 2011; Ipeirotis and Gabrilovich 2014; Gao et al. 2011). The crowdsourcing model (Trottier 2010; Surowiecki 2004), although novel in some important respects, encapsulates a key question that has captured the attention of economists, psychologists, and operations research researchers for a long time—whether and how financial incentives can be used to motivate workplace performance (Mason and Watts 2010; DiPalantino and Vojnovic 2009; Finnerty et al. 2013; Kittur, Chi, and Suh 2008; Yin, Chen, and Sun 2014). Indeed, workers sometimes are willing to work for free, motivated by intrinsic incentive such as enjoyment, curiosity for new knowledge (Kaufmann, Schulze, and Veit 2011) or interests in contributing to a certain scientific discipline (Mao et al. 2013). However, in most general crowd-sourcing platforms such as Amazon Mechanical Turk, the primary type of incentive remains the extrinsic motivation of monetary compensation (Yin and Chen 2015). Thus the key challenge for the crowdsourcing system designer is whether and when to offer such rewards in a working session in order to maximize utility.

One important question that was investigated in this context is the correlation (or its absence) between the payment for a task and the effort workers put in that task. Generously rewarding workers upon completing the task was found inadequate in ensuring that workers will pay much attention while engaged with the task. In fact it has been shown that even at low compensation rates, payment levels do not appear to affect work quality but only the quantity of tasks workers complete (Mason and Watts 2010). Moreover, it was shown that financial incentives can be used to trade quality for speed (Mao et al. 2013). The alternative for a fixed payment upon completion of a task is rewarding based on performance, which is also the approach used in this paper. Here, various variants have been studied. For example, workers can be rewarded based on "relative performance" (also known as "crowd-sourcing contests" (Gao et al. 2012; Feng et al. 2014)). Namely only the worker that performed best receives rewards. Another example is rewarding for "absolute satisfying performance" based on pre-specified criteria (Ho et al. 2015). According to this method each worker that meets the pre-specified criteria, receives the reward regardless of the performance of the other workers. Within this context Yin and Chen (2015) have suggested methods to intelligently decide whether and when to offer such rewards to maximize the overall utility. Their claim was that since the performance-contingent rewards can affect the quality of the crowd's work, it is not necessarily always beneficial to provide such rewards, as the potentially improved quality comes with an increase in cost.

One important characteristic of the monitoring task considered in this paper is that it is monotonous and somehow boring. Yin et al. (2014) dealt with the problem of a task switching setting within a task series to diversify the work or to avoid fatigue or boredom within a crowdsourcing platform. They found that their monetary intervention scheme that rewards a subset of tasks within a series of tasks is effective in improving work quality by the spillover effect on the non-intervened tasks in the session, in terms of response time and accuracy. Their solution, however, is inapplicable in our case, as we only have a single type task and cannot switch in-between. Another interesting work dealing with boredom within a crowdsourcing platform (Dai et al. 2015) suggested that diversions can help to retain workers attention. However, the use of diversions was not suggested as a mechanism for improving the work quality.

Another relevant line of work considers the problem of monitoring those who monitor using a game-theoretic ap-

proach. For example, Rahman (2012) suggests adding a higher level or monitoring task (Rahman 2012). The difference from our work is that it considers fully rational players, while we consider human workers that are known to act irrationally (Hajaj, Hazon, and Sarne 2015; 2016; Elmalech, Sarne, and Agmon 2016; Elmalech, Sarne, and Grosz 2015; Elmalech et al. 2015).

Finally, some may find some resemblance between our idea of dynamically generating dummy events in order to incentivize workers to focus in the task and the idea of reward shaping used in reinforcement learning techniques (Laud 2004; Tenorio-Gonzalez, Morales, and Villaseñor-Pineda 2010). Reward shaping technique help the learning agent or multi-agent system to develop association between an action and its effectiveness towards the goal completion. This way, the learning is achieved in a much faster way than the classic reinforcement learning that mainly faces the challenges of navigating in a huge state space with no information or using substantially large amount of experience of the given domain (Devlin, Kudenko, and Grześ 2011; Babes, De Cote, and Littman 2008). While the use of reward shaping in reinforcement learning techniques is for accelerating the learning convergence, in our work the use of the reward scheme is for maintaining the worker's attention through the monotonous task she is engaged in.

Model

We consider settings where a monotonous task needs to be outsourced to workers. The task requires moderate, yet continuous, attention on the worker's side, with a very low cognitive load. The measure of success for the worker is fully based on paying attention to and identifying a well-defined event of interest. The main characteristics of such an event are: (i) it is rare, and (ii) it may be clearly defined for a human but will be very ambiguous for a machine or for a computer system. Typical tasks of the above nature include monitoring (e.g., keeping track of a child playing in a playground), surveillance (e.g., monitoring a camera at a public park to alert in case of a crime) and reading content to ensure it does not contain offensive words. In all these tasks, there is a very small probability for the occurrence of one or more events of a specific type, and the worker's success in executing the task exclusively depends on whether the event was identified (if occurred).

Formally, we define the task as the identification of a well-defined event E of duration E_t starting at time t within time interval $(0, T > t)$ in which other various events different than E take place. The worker controls at all times whether her attention is placed on the outsourced task or other tasks (that are irrelevant to the outsourced task). The worker is able to identify the event E only if her attention is placed on the task within the interval $(t, t + E_t)$.

We emphasize that the above model fits both online and offline monitoring tasks. Taking the CCTV domain as an example, here an event can be the appearance of survivors from a plane crash at sea in aerial videography (offline) or the identification of theft in a convenience store (online). It is very difficult to train a computer vision system to distinguish between an honest buyer who is taking an item from the shelf for legitimate reasons, and a dishonest customer intending to steal it, or between airplane and equipment remains and survivors (at sea).

As with most crowdsourcing platforms, our model assumes workers can be rewarded based on well-defined criteria (e.g., for identifying the event of interest). Therefore, the performance measures for mechanism aiming to incentivize workers in applications of the above types is a combination of the expected expense on a worker and the probability the worker will identify the event of interest. In particular, we are interested in the tradeoff between the two, meaning that one method will be considered better than another if it can provides a better detection probability for the same expected expense as the other or, alternatively, if it can achieve the same detection probability for a lesser expected expense.

Generating Dummy Events

Our proposed method for improving workers' performance in tasks that requires the worker's attention relies on artificially embedding dummy events along the task, in an intelligent controlled manner. A dummy event is an event of interest for which the worker is compensated if identified on time, despite the fact that its identification is useless for the task allocator. There are three main motivations for our use of dummy events. First, dummy events enable the fully attentive worker to accumulate rewards throughout the task, rather than waiting for a single meaningful event which identification (or lack of) completely governs the payment to be received. This reduces the variance in the payment received and eliminates the risk of not receiving a payment at all due to a possibly single occurrence of not paying attention to the task for a very short period of time. These last two properties are highly desirable when dealing with human workers(Kahneman and Tversky 1979). Second, with the introduction of dummy events the task as a whole becomes more interesting to the worker. Instead of simply staring at the screen for a relatively long time, waiting for a meaningful event to occur, the worker is now more active and vibrant throughout the task. Finally, and perhaps most important, the worker's success in spotting any dummy event can be used as an indication for her attentional state at that specific time. In particular, a failure to identify a dummy event can indicate that the worker is now busy doing something else (and any consecutive failure is likely to strengthen this belief).

The introduction of dummy events throughout a task is challenging in the sense that there are several parameters affecting its effectiveness. An intelligent design of a dummy-events based mechanism should properly determine the number of dummy events to introduce to workers, the timing of the insertion of the dummy events along the task, the payment for each successful identification of a dummy event and the payment for the primary event of interest. This, as opposed to only a single parameter that needs to by tuned when not using dummy events (i.e., the payment for the primary event of interest). Naturally, the questions relating to the magnitude of the payment for properly identifying different event types can be resolved experimentally, i.e., through trying different combinations of payments, extrapolating the expected accuracy in identifying the primary event of interest as a function

of the payments made. In this paper, we therefore focus on the mechanisms for timing the insertion of the dummy events.

Two intuitive methods for generating dummy events along the task are to spread the events evenly (uniformly) along the task and to randomly draw the timings when such events should appear. Both methods guarantee that the dummy events are spread along the task to avoid long periods of time with no dummy events (that can potentially bore the worker and push her to abandon the task or temporarily focus in something else). The uniform spread guarantees a more steady flow of dummy event, hence has the benefit of convincing the worker that it is beneficial to keep focused in the task. It does, however, have a drawback in the sense that the worker may quickly learn the dummy-event generation pattern and consequently switch to other tasks in-between, as she knows when the next dummy event will appear. Furthermore, once recognizing the dummy-event generation pattern the worker will be able to assess the expected payment resulting from detecting the dummy events. Since the goal at the end of the day is to increase the attention span with a lower effective payment overall, the worker, realizing it is not rewarding enough, is likely to become disappointed and abandon the task. The second method that uses random timings resolves this latter problem, however can lead to relatively long periods of time where no dummy event is presented to the worker.

In addition to the above methods, we propose a third mechanism that does not determine the exact times for introducing dummy events a priori. Instead it dynamically allocates dummy events in a way that motivates workers to stay tuned to the task (hence we refer to it onwards as DDEA - Dynamic Dummy-Event Allocation). The decision to introduce a dummy event at any given time is probabilistic, where the probability of such event depends on the time elapsed since the last introduction of a dummy event and to some extent also on the weighted aggregated prior behaviors exhibited by the worker, as captured by the results (success or failure in identification) of formerly introduced dummy events. These two factors influence the value of an attentiveness measure that the mechanism maintains, denoted F, aiming to capture the worker's attentional state at each time. The value of F ranges between $0-1$, where 0 indicates no attention to the task and 1 represents full attention. Failing to identify a dummy event will result in a relatively sharp decrease in the value of F, whereas a correct identification will result in a sharp increase. Both the increase and the decrease in the value of F use exponential smoothing techniques such that the new value depends on all prior values of the measure, with an exponentially decreasing weight to each prior value according to the time elapsed since it was determined. Additional adaptations to the value of F occur based on the time elapsed since the last evidence of the worker's attentional state was received, i.e., based on the time that elapsed since the last sub-task was introduced, in a way that exhausts the value of F over time. The decrease of the value of F over time reflects to some extent typical people's attention span model (e.g., Figure 6.4, page 60, in (Aarabi 2007)).

The adaptation process is compactly captured in Algorithm 1. The value of F is first initialized to 1, as it is most likely that at the beginning of the task the worker is fully tuned to it. The decision concerning the introduction of a dummy event takes place every few seconds (modeled using the parameter $DecisionPointsInterval$). Ideally the value of this parameter should be infinitesimal, for reaching as many dummy event decision points along the task. Still, in order to avoid a continuously running thread, one may prefer to use some non-negligible time interval between decision points. Once a new decision point is reached (Step 2), the mechanism reduces F by a factor of δ. This corresponds to the probability that the worker is not in focus (i.e., doing something else), which increases as time goes by and no other indication of her attentiveness was received. The choice of the proper δ value depends on the variable $DecisionPointsInterval$—the greater the value of $DecisionPointsInterval$, i.e., the greater the time elapsed since the last time the value of F was reduced, the greater the reduction in F should be. For example, the discounting of F from 1 to, say, third, whenever discounting every 3 seconds ($DecisionPointsInterval = 3sec$) requires $\delta = 0.99$, if discounting over 5 minutes and $\delta = 0.997$ if discounting over 20 minutes. These of course refer to the case where the value is continuously discounted, without receiving any new information from the introduction of a dummy event.

The F value is then used for deciding whether or not to introduce a dummy event (Step 4). This is achieved by comparing F to a random number drawn from a uniform probability distribution function in the range $0-1$. In case a dummy event is introduced and identified by the worker (Step 7), the value of F is increased. The increase has a fixed component, represented by α and the remaining increase is positively correlated with the current value of F. Suggested values for α are thus within the range of $0.80-0.95$,[2] representing a relatively high confidence in having the worker's attention fully focused in the task based on a successful identification of a dummy event. In case the dummy event was not identified upon its introduction to the worker, the value of F decreases by a factor of β (Step 10). The value of β should be substantially smaller than δ (in at least one order of magnitude), as the events are very different—while β corresponds to the event of a dummy event introduced to the worker and not properly identified, δ corresponds simply to the increased chance of losing focus as time goes by. Still, the idea is that the values for β will not be too small (e.g., such that F will become too close to zero), because it is possible that the worker is generally tuned to the task, but due to a temporary disturbance missed the dummy event.

Naturally, we expect the third method to outperform the first two, as unlike them it correlates the choice of introducing a dummy event with some prediction of the worker's current attentional state. Still, we believe it is important to study the first two methods due to their simplicity, intuitiveness and the fact they put a bound on the total expense.

[2]Using $\alpha = 1$ is not recommended, as we do not want the measure F to trivially jump to 1. It is possible that the worker noticed the dummy event in the corner of her eye or that she accidentally ran into the application by switching tasks at the time the dummy event emerged.

Algorithm 1: Dynamic Dummy-Event Allocation (DDEA).

input : $DecisionPointsInterval$

initialization:
$NextDecisionPoint = CurrentTime()$;$F = 1$; **while** *TaskIsOn* **do**
  if $CurrentTime \geq NextDecisionPoint$ **then**
    $F = F * \delta$;
    if $Random() < F$ **then**
      IntroduceDummyEvent();
      NextDecisionPoint+=DecisionPointsInterval;
      if *worker identified DummyEvent* **then**
        $F = \alpha + (1 - \alpha)F$;
      else
        $F = \beta F$;
      end
    end
  end
end

Experimental Infrastructure

For the experiments we used an Internet game called "Find the Animal". The game's GUI is composed of four tiles visible to the worker, each with a different picture from a repository of 45 cartoon animals. Figure 1 presents a screen-shot of this game. Each k second the picture on one of the tiles is replaced by a different one from the repository, where both the tile that will be changed and the new picture are chosen randomly. The worker gains rewards in the game whenever clicking on a tile that has one of some pre-specified pictures appearing on it. The worker receives a graphical indication (a summary of the number of missed pictures of interest, appearing at the bottom of the screen) for every event that was not identified on time. The game enables the experiment administrator to set the time or times a specific animal will appear (with no control over the specific tile it will appear on) as well as full control over the code deciding when to introduce a specific animal on a tile. The first functionality is aimed to support the introduction of the main event/s of interest while the second is used to support the introduction of dummy events. The game was implemented as a web-based application that uses a relatively simple graphical interface, to facilitate interaction with workers using almost any screen size.

The "Find the animal" game is a good representation of our problem domain: it contains a stream of various types of events continuously occurring along the task (represented by the changes in the pictures on the four tiles), the event of interest and dummy events can be trivially defined as the appearance of specific pictures, an event lasts a short period of time, the task of identifying a specific picture is easy to understand and execute (if focusing in the task) and does not require any special skills. A person can succeed in identifying the event of interest, captured by the appearance of a specific animal picture, only by paying attention to what is displayed on the tiles. The animal pictures are very different and therefore if paying attention to the tiles the identification of the event is immediate. The game is easy to understand

Figure 1: A screen-shot of the game.

and lacks any complexities or external factors that may affect the results (other than the worker's focus, which is the investigated parameter) hence enables direct measure of the effectiveness of the tested methods. In terms of application domain the game can be mapped to many real-life tasks, e.g., the identification of a specific person (or a group of people or even a specific car) in a crowd (e.g., among pedestrians passing a street or cars driving in a traffic lane). Indeed there are some real world tasks that can be considered less or more tiring (in terms of the cognitive load exerted in our game). Still, the environment complexity and the cognitive load exerted are secondary in their effect over the workers' success, thus the game is a good fit for testing our hypothesis.

Experimental Design

We configured the "Find the Animal" game for our purposes such that every ~ 4 seconds a different animal picture was changed (i.e., $k = 4$). The length of the game was set to 40 minutes.[3] The event-of-interest picture was set to be a duck (hence the game was named "Find the Duck" when uploading it to AMT) and the time for the duck to appear in each game was randomly drawn at the beginning of the game. In the dummy event configuration of the game, the worker was credited also for clicking on a tile with a goat picture.

Each participant received thorough instructions of the game rules and her goal in the game. The instructions emphasized that only participants that keep the application running until the end of the game (i.e., for the entire 40 minutes) will get paid for the HIT. Participants received an explanation about the compensation structure, which was composed of a show-up fee (fixed wage) of 5¢[4] and a bonus which depended on whether or not the event of interest was identified and the number of dummy events spotted by the participant throughout the experiment. The information regarding the events that was provided to participants specified that one duck will appear at some unknown time (equivalent to set-

[3] Prior work presents general evidence for teenagers and adults inability to sustain attention on one thing for more than about 40 minutes at a time (David Cornish, Dukette, and others 2009).

[4] AMT requires that employers pay workers at least 1¢ (basic payment) for completing a task.

tings where we know a crime event happened throughout the task, yet the exact time is unknown) whereas the number of goats that will appear (for the treatments where the dummy-event methods were tested) remained unknown. After the instructions step, participants were asked to answer a short quiz in order to ensure that they truly understood the game rules and their goal in the game. Upon completing the 40 minutes task participants were informed of the bonus based on their performance.

Participants were recruited and interacted through Amazon Mechanical Turk (AMT) and were assigned to one of 19 game sessions of 4 treatments as specified in the following table:

Treatment	Duck (primary)	Goats (dummy)
No dummies (X8)	$bonus \in \{0¢, 10¢, 20¢, 40¢, 60¢, 80¢, 100¢, 200¢\}$	N/A
Random (X5)	$bonus = 10¢$	$bonus = 1¢$, # of goats$\in \{10, 20, 30, 40, 50\}$
Uniform (X5)	$bonus = 10¢$	$bonus = 1¢$, # of goats$\in \{10, 20, 30, 40, 50\}$
DDEA (X1)	$bonus = 10¢$	$bonus = 1¢$, # of goats=according to Algorithm 1

In the first treatment ("No dummies") we aimed to test the method of controlling the worker's attentional state through the reward she receives for identifying the primary event of interest. Hence no goats were used and the bonus promised exclusively depended on whether or not the duck was found. We had 8 variants of this treatment, differing in the bonus awarded for finding the duck: 0,10,20,40,60,80, 100 and 200 cents. The other three treatments aim to test the dummy-events based methods: generating dummy events in uniform intervals, at random times and using Algorithm 1. In all three the workers were promised a $10¢$ bonus for spotting the duck and $1¢$ for each goat (dummy event). The treatments "Random" and "Uniform" test generating dummy events at random and uniform times, respectively. Each such treatment was used with five variants, differing in the number of goats used $(10, 20, 30, 40, 50)$. The treatment DDEA used the method of dynamic dummy event augmentation according to Algorithm 1. The values set for the different parameters were: $\delta = 0.999$, $\alpha = 0.9$ and $\beta = 0.4$, for the justifications provided when introducing the algorithm. We note that the idea was to provide a proof of concept for the effectiveness of the proposed approach, rather than to find the optimal configuration for the "Find the animal" game. Consequently we did not check additional configurations, as even with these (possibly sub-optimal) parameters the method was found to perform better than all other method, even when the latter were configured optimally.

Overall, there were 30 different participants for each game session (570 participants overall), differing in age (21-60) and gender (51% men and 49% women). In each game session the maximal gender ratio difference was 6%. We didnt employ any selection criteria for recruiting Turkers, as we attempt to emulate a crowdsourcing scenario where you need to recruit a large pool of workers in a very short time. To prevent the carryover effect, a "between subjects" design was used, i.e., for each session a different group of participants was recruited. For each participant we logged the times that a duck/goat appeared and if the participant clicked or missed the goat/duck.

Results

No dummies

Figure 2 depicts the required bonus and the resulting expected effective payment as a function of the detection probability one aims to achieve when not using dummy events (the "No-dummies" treatment).[5] The first curve is based on the results of the eight "No-dummies" treatment variants. Each of its data points represents the appropriate percentage of participants who managed to spot the duck in the appropriate session (horizontal axis) for a specific value tested as the bonus for finding the duck (vertical axis). The curve was smoothed by means of the *smoothing spline* method,[6] resulting in $f(x) = 1575 * x^3 - 1271 * x^2 + 401.6 * x - 39.27$ (with $R^2 = 0.992$). The second curve which represents the expected actual payment required for guaranteeing the detection probability of the horizontal axis is a direct transformation of the latter curve. The transformation is done by multiplying the proposed bonus by the probability it will actually be awarded (i.e., the probability on the horizontal axis) and adding the $5¢$ fixed payment for the HIT.

The shape of the curve is quite obvious - as expected, the probability that a worker will identify the primary event of interest increases as the reward promised for spotting it increases. The importance of the graph, however, is in enabling the extraction of a baseline for comparison, as the exact marginal improvement due to any additional cent promised as a bonus can only be found through experimentation and smoothing. The curve given in the figure is concave, indicating that the bonus increase required to achieve an increase in the detection probability increases as the detection probability increases. Interestingly, a bonus of $\$2$ obtains a detection probability of 74% and in order to guarantee detection (a 1-ϵ detection probability), one needs to offer a bonus of $\$6.7$, based on the smoothed curve. Note that in this latter case the offered bonus is also the expected (and actual) effective bonus, as it is necessarily paid to the worker. This latter finding makes sense, as the average wage of AMT workers was reported to be $\$4.8$ (Ipeirotis 2010), hence the typical worker would expect a payment of slightly more than that in order to keep focused in our game for 40 minutes.

Based on the tradeoff between the expected payment and the achieved precision encapsulated in Figure 2 one can

[5] While flipping the axes may seem more intuitive (as the promised bonus is the independent variable), we prefer the current presentation as it better captures the tradeoff between the achieved detection probability and expense. Furthermore, only this presentation method supports the inclusion of both parameters (bonus and expected payment) in one graph.

[6] The smoothing spline is a method of fitting a smooth curve to a set of noisy observations using a spline function.

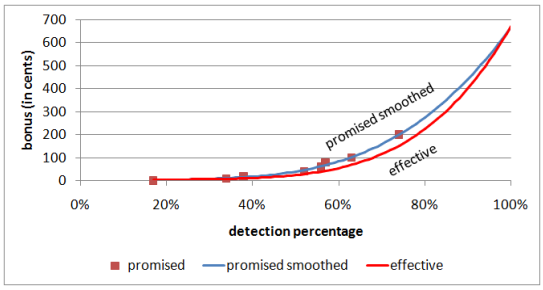

Figure 2: The correlation between detection percentage, reward promised and effective payment.

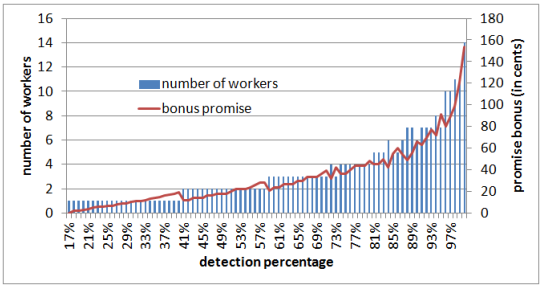

Figure 3: The optimal number of people the task will be distributed to and the average bonus promised as a function of the detection percentage.

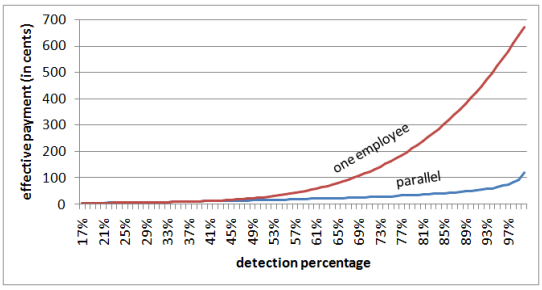

Figure 4: The average effective payment for one employee compared to a number of employees in parallel as a function of the detection percentage.

choose a preferred working point and set the bonus accordingly. Furthermore, Figure 2 enables designing an even more cost-effective detection process through employing parallelism in the process. The idea in parallelizing the process is to recruit several workers for the task, having each working (and receiving payments) independently of the others and their performance. While having more workers identify the event of interest results in redundant bonus payments, the parallel effort enables offering smaller bonuses in the first place and with proper design can actually reduce the overall expected expense required for each pre-defined detection probability set by the system manager. We use $P(x)$ to denote the detection probability achieved by promising an event-detection reward x to a random worker. The value $P(x)$ can be extracted from the extrapolated graph given in Figure 2, i.e., given by $P(x) = f^{-1}(x)$. The detection probability when assigning the task to N workers in parallel, while offering a success-based reward of x, denoted $P_N(x)$ is given by: $P_N(x) = 1 - (1 - P(x))^N$ and the expected expense in this case is $N \cdot (HIT + b(x))$, where HIT is the base payment (for participating) and $b(x)$ is the effective bonus payment corresponding to an offered bonus x as given in Figure 2.

Therefore, based on the effective bonus curve given in Figure 2 we can find for any requested detection probability the expected-expense-minimizing pair (x, N).[7] This is illustrated in Figure 3 which depicts the optimal (expected-payment-minimizing) number of workers and the bonus that needs to be offered to them for spotting the primary event, for each required detection level (horizontal axis), respectively. Figure 4 summarizes the expected overall effective expense (including the $5¢$ base payment for the HIT) for each required detection level, for the case of using parallel detection and the case of using a single worker. The first curve is identical to the one given in Figure 2, though shifted vertically by the $5¢$ fixed payment for the hit (which is a bit difficult to observe because of the resolution of the figure). The second derives from Figure 3, taking into consideration both the number of workers used and the effective payment made to each. Figure 4 demonstrates that the reduction in the expected payment that can be achieved through the use of parallel detection is substantial, and increases as the required detection level increases. This latter phenomena is explained by the concavity of the curves in Figure 2—the substantial decrease in the individual effective bonus enables recruiting several workers for the same amount of expected expense, while the parallel detection fully outweighs the resulting minor decrease in the individual detection probability.

[7]We use a fixed bonus for all N workers, since on AMT the same bonus must be offered for the same task (HIT). Otherwise, we could have gotten slightly better results by using a mixture of different bonus offers to different workers.

Random and uniform dummy events generation

Next, we report the results obtained with the use of dummy events when using random and uniform generation patterns. Recall that in the game sessions in which dummy events were introduced we offered workers a bonus of $10¢$ for finding the duck and $1¢$ for each dummy event (goat). The resulted detection probability and the corresponding effective expense for each of the uniform and random dummy event generation method variants are given in Table 1.

From Table 1 we observe that, as expected, the increase in the number of dummy events introduced into the task results in an increase in the detection probability, with the cost of an increase in the effective expense. Neither method (uniform or random dummy event generation) generally dominates the other and the detection probability is almost identical with the two for any number of dummy events tested. Similarly, the effective expense is somehow similar, with the most notable difference when having 50 dummy events overall. Here, the expected expense with the random method is lower, possibly

	random		uniform	
	effective expense	detection percentage	effective expense	detection percentage
10 goats	17.6	67%	18.4	67%
20 goats	24	70%	27.3	70%
30 goats	39	80%	37	81%
40 goats	49.8	83%	48.3	81%
50 goats	52.9	83%	58.4	83%

Table 1: Detection percentage and effective cost as a function of number of goats.

explained by people's ability to learn the pattern of generating dummy events with the other method. Still, one would expect a similar learning effect with the 30 and 40 treatment variants, which is not the case.

In comparison to the "no-dummies" approach, both the random and uniform dummy event generation methods suggest a substantially more competitive tradeoff—a detection probability of $67\% - 83\%$ is achieved with an effective expense of $17.6¢ - 49.8¢$ and $18.4¢ - 58.4¢$ for random and uniform, respectively, compared to a required effective expense of $94¢ - 269.7¢$ with "no dummies" for this interval of detection probabilities. These differences are statistically significant using t-test ($p < 0.01$).[8]

In an effort to reason about the behavior of workers, and in particular their attention span throughout the task, we present Figures 5 and 6. The figures present the distribution of workers in the sampled population according to the percentage of dummy events they detected out of the total number of such events that were presented to them (over all 5 treatments in each method). From the figures we observe that indeed a large portion of the population found the generation rate of the 10 and 20 goats treatments unappealing, resulting in a lower than 50% dummy event detection, which can be interpreted (with high level of confidence, due to the nature of the two methods) as losing focus over a major portion of the task. With 40 dummy events and more (and in the case of random allocation with 30 dummy events and more) most workers stayed focused for almost the entire duration of the task (80-100% of the time). The relatively higher goat detection percentage values in the case of the uniform allocation method compared to the random method can be attributed to the constant rates that the dummy events appeared in the first: it is possible that workers managed to learn the pattern according which dummy events appear and became more responsive at times they anticipated a dummy event to appear.

Figure 7 presents the correlation between the percentage of dummy events identified by workers throughout the task (horizontal axis) and the detection probability achieved by these workers. Each bar summarizes the results for all five

[8] In the absence of data related to the exact same detection probability in all treatments, we compared those treatment variants that achieved at least the same detection probability as with the compared "no-dummy" treatment variant, in order to carry out the t-test. The comparisons made were thus between the 10 goats and the 30 goats treatment variants of the uniform and the random treatments to the 100¢ and 200¢ treatment variants of the "no-dummy" treatment.

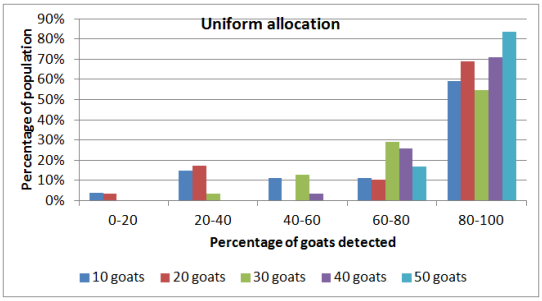

Figure 5: Uniform generation: the distribution of the population according to the percentage of dummy events identified, for different number of generated dummy events.

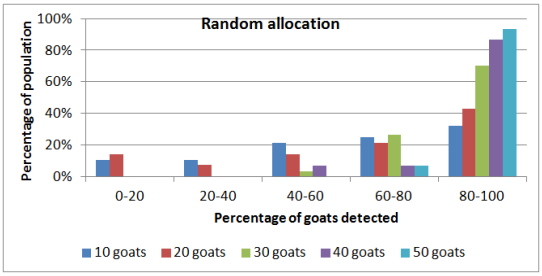

Figure 6: Random generation: the distribution of the population according to the percentage of dummy events identified, for different number of generated dummy events.

treatment variants. As expected, there is a strong correlation between the percentage of dummy events identified by workers and their detection probability (the calculated correlation coefficients are 0.93 and 0.97 for the random and uniform dummy event generation methods, respectively). The correlation between the two variables enables the system owner some sort of a retrospect validation in case the event of interest is not identified: if the number of dummy events identified by the worker in the task is high then it is most likely that the reason for not identifying the event of interest is that it actually did not occur (e.g., when monitoring a stream of video offline, attempting to find out if a suspect passed through the specific location of the camera while escaping). If, on the other hand, the number of dummy events identified by the worker in the task is low then it is most likely that the worker did not pay attention to the task, and an additional worker should be assigned to it.

DDEA dummy events generation

Finally, our DDEA-based method achieved a detection probability of 81% with a corresponding effective expense of 24¢. This result is substantially better than those obtained with the first two methods—from Table 1 we observe that the effective expense required for achieving this level of precision dictates an expected expense of somewhere between $39 - 49.8¢$ with the method that generates dummy events at random times and 37¢ with the one that spreads them uniformly.[9] The differ-

[9] Alternatively, for a payment of 24¢ one can achieve a detection probability of 70% with the random method and somewhere

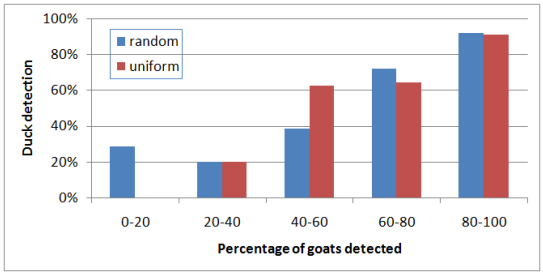

Figure 7: The detection probability (of the event of interest) as a function of the percentage of dummy events detected.

ence in the expected expense is statistically significant using t-test (taking the 81% detection probability as a baseline). Compared to the method that relies solely on rewarding the identification of the event of interest ("no-dummies"), the performance of DDEA is strikingly better—the corresponding expected expense for assuring a detection probability of 81% is $239¢$ (10 times more!) and the detection probability achieved in exchange for an effective expense of $24¢$ is 49%, according to Figure 2. Furthermore, even compared to the use of parallel detection, as suggested above, the use of our intelligent dummy event insertion method achieves better results (even if used with a single worker) - the 81% precision detection is achieved with an expected expense of $24¢$ compared to $38¢$ which is the minimum achievable when using parallel detection according to Figure 4.

Conclusions

Methods for motivating workers when using crowdsourcing for application involving the monitoring of rare events are likely to focus much interest in future years, as these applications become ubiquitous. The exceptional growth in deploying CCTV surveillance cameras as well as various other sensors suggests an increase in the demand for workers and paves the way for outsourcing monitoring tasks in new application domains. The recent advances in Internet technology as well as the wide spread of mobile technologies suggest that the pool of potential ad-hoc workers for such tasks is likely to increase immensely in coming years. The monitoring task is simple and requires only the worker's attention rather than any special skill hence ideal for crowdsourcing.

The encouraging results reported in the former section support our hypothesis that the effective way for increasing workers' attention span in such monotonous crowdsourcing monitoring tasks is the one that uses dummy events rather than the traditional method of increasing the reward for identifying the primary event of interest. As discussed throughout the paper, the benefits of the method are threefold: it enables workers to accumulate rewards throughout the task, rather than waiting for a single meaningful event, hence reducing the variance in the payment received, the task as a whole may become more interesting to the worker and it enables indications for the worker's attentional state throughout the task. There are many ways for introducing dummy events in various domains. For example using augmented reality tools (as in *www.clandestineanomaly.com*). These can be either domain-related (e.g., introducing virtual penetrators in security applications) or unrelated (e.g., inserting goats), as the purpose is merely to keep workers tuned to the application. Naturally, the transition to dummy-events-based design calls for a method for timing the dummy events throughout the task. In this paper we proposed and evaluated three such methods. All three were found to be substantially more effective compared to the traditional method of exclusively correlating the reward with the identification of the event of interest, offering a better tradeoff between the detection probability and the required expected expense. The advantage of the first two methods is in their simplicity and in the fact that they bound the number of dummy events to be presented to the worker (hence also the actual expense). The third method generates dummy events dynamically. This method was found to perform best out of the three, demonstrating the effectiveness of its core algorithm.

We emphasize that the results provided for our methods are actually lower bounds for the performance one may achieve if using them properly. The baseline method that relies on rewarding only based on identifying the event of interest was evaluated with a wide range of rewards, enabling the extraction of a curve fully capturing the impact of the proposed reward over the detection probability and the expected expense. Therefore this method can be fully tuned. Our two methods that rely on uniform and random generation of dummy events were evaluated based on varying the number of dummy events however the reward for correctly identifying the event of interest and dummy events was arbitrarily set. Finally, for our dynamic method, all parameters (payments for the primary and dummy events, F, $NextDecisionPoint$ and δ) were all arbitrarily set, since this was merely a proof of concept. Therefore, with the right tuning (possibly with further experimentation) we expect our methods to perform even better compared to the baseline, and in particular we expect the improvement in our dynamic method (DDEA) compared to the baseline and to the other two methods to be the greatest.

We see many directions for future research emerging from this paper, out of which we detail three. First, as mentioned above, much work is still needed for finding good ways for tuning the proposed methods. The tuning process, of course, is application-dependent, and therefore we chose not to include it within the scope of the paper, especially given that the results obtained even with the arbitrary parameters that we used demonstrated a substantial improvement. Still, coming up with general tuning methodologies for such methods can be highly beneficial. Second, we believe that better performance can be achieved by modeling workers and tailoring the parameters to each worker individually. One option for user modeling in this case is based on the signals received by the identification or misidentification of the initial dummy events or any governing pattern observed. Finally, we propose to combine the decisions on generating dummy events with other measurable factors such as the cursor position on the screen (e.g., within the application window or not) and the status of the application window (active or partially

between 67% and 70% when using the uniform method.

covered by another window).

Acknowledgement

This research was partially supported by the ISRAEL SCIENCE FOUNDATION (grant No. 1083/13) and the ISF-NSFC joint research program (grant No. 2240/15).

References

Aarabi, P. 2007. *The art of lecturing: a practical guide to successful university lectures and business presentations*. Cambridge University Press.

Babes, M.; De Cote, E. M.; and Littman, M. L. 2008. Social reward shaping in the prisoner's dilemma. In *Proc. of AAMAS*, 1389–1392.

Bernstein, M. S.; Brandt, J.; Miller, R. C.; and Karger, D. R. 2011. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proc. of UIST*, 33–42.

Bernstein, M. S.; Karger, D. R.; Miller, R. C.; and Brandt, J. 2012. Analytic methods for optimizing realtime crowdsourcing. *arXiv preprint arXiv:1204.2995*.

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

David Cornish, M.; Dukette, D.; et al. 2009. *The essential 20: Twenty components of an excellent health care team*. Dorrance Publishing.

Devlin, S.; Kudenko, D.; and Grześ, M. 2011. An empirical study of potential-based reward shaping and advice in complex, multi-agent systems. *Advances in Complex Systems* 14(02):251–278.

DiPalantino, D., and Vojnovic, M. 2009. Crowdsourcing and all-pay auctions. In *Proc. of ACM-EC*, 119–128.

Doan, A.; Ramakrishnan, R.; and Halevy, A. Y. 2011. Crowdsourcing systems on the world-wide web. *Communications of the ACM* 54(4):86–96.

Elmalech, A.; Sarne, D.; Rosenfeld, A.; and Erez, E. S. 2015. When suboptimal rules. In *Proc. of AAAI*, 1313–1319.

Elmalech, A.; Sarne, D.; and Agmon, N. 2016. Agent development as a strategy shaper. *Autonomous Agents and Multi-Agent Systems* 30(3):506–525.

Elmalech, A.; Sarne, D.; and Grosz, B. J. 2015. Problem restructuring for better decision making in recurring decision situations. *Autonomous Agents and Multi-Agent Systems* 29(1):1–39.

Feng, Z.; Zhu, Y.; Zhang, Q.; Ni, L. M.; and Vasilakos, A. V. 2014. Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In *Proceedings of INFOCOM*, 1231–1239.

Finnerty, A.; Kucherbaev, P.; Tranquillini, S.; and Convertino, G. 2013. Keep it simple: Reward and task design in crowdsourcing. In *Proc. of SIGCHI*, 14:1–14:4.

Franklin, M. J.; Kossmann, D.; Kraska, T.; Ramesh, S.; and Xin, R. 2011. Crowddb: answering queries with crowdsourcing. In *Proc. of ACM-SIGMOD*, 61–72.

Gao, H.; Wang, X.; Barbier, G.; and Liu, H. 2011. Promoting coordination for disaster relief–from crowdsourcing to coordination. In *Social computing, behavioral-cultural modeling and prediction*. Springer. 197–204.

Gao, X. A.; Bachrach, Y.; Key, P.; and Graepel, T. 2012. Quality expectation-variance tradeoffs in crowdsourcing contests. In *Proc. of AAAI*.

Hajaj, C.; Hazon, N.; and Sarne, D. 2015. Improving comparison shopping agents' competence through selective price disclosure. *Electronic Commerce Research and Applications* 14(6):563–581.

Hajaj, C.; Hazon, N.; and Sarne, D. 2016. Enhancing comparison shopping agents through ordering and gradual information disclosure. *to appear in JAAMAS*.

Ho, C.-J.; Slivkins, A.; Suri, S.; and Vaughan, J. W. 2015. Incentivizing high quality crowdwork. In *Proc. of WWW*, 419–429.

Ipeirotis, P. G., and Gabrilovich, E. 2014. Quizz: Targeted crowdsourcing with a billion (potential) users. In *Proc. of WWW*, 143–154.

Ipeirotis, P. G. 2010. Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students* 17(2):16–21.

Kahneman, D., and Tversky, A. 1979. Prospect theory: An analysis of decision under risk. *Econometrica: Journal of the Econometric Society* 263–291.

Kaufmann, N.; Schulze, T.; and Veit, D. 2011. More than fun and money. worker motivation in crowdsourcing-a study on mechanical turk. In *AMCIS*, volume 11, 1–11.

Kittur, A.; Chi, E. H.; and Suh, B. 2008. Crowdsourcing user studies with mechanical turk. In *Proc. of CHI*, 453–456.

Laud, A. D. 2004. *Theory and application of reward shaping in reinforcement learning*. Ph.D. Dissertation, University of Illinois at Urbana-Champaign.

Levy, P., and Sarne, D. 2016. Intelligent advice provisioning for repeated interaction. In *Proc. of AAAI*, 842–849.

Loomes, G., and Sugden, R. 1982. Regret theory: An alternative theory of rational choice under uncertainty. *The economic journal* 92(368):805–824.

Mao, A.; Kamar, E.; Chen, Y.; Horvitz, E.; Schwamb, M. E.; Lintott, C. J.; and Smith, A. M. 2013. Volunteering versus work for pay: Incentives and tradeoffs in crowdsourcing. In *Proc. of HCOMP*.

Mason, W., and Watts, D. J. 2010. Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter* 11(2):100–108.

Norris, C.; McCahill, M.; and Wood, D. 2004. Editorial. the growth of cctv: A global perspective on the international diffusion of video surveillance in publicly accessible space. *Surveillance & Society* 2(2/3):110–135.

Rahman, D. 2012. But who will monitor the monitor? *The American Economic Review* 2767–2797.

Surowiecki, J. 2004. *The Wisdom of Crowds*. New York Anchor Books.

Tenorio-Gonzalez, A. C.; Morales, E. F.; and Villaseñor-Pineda, L. 2010. *Dynamic Reward Shaping: Training a Robot by Voice*. Proc. of IBERAMIA. 483–492.

Trottier, D. 2010. An inventory and evaluation of cctv internet crowd-sourcing. *The Privacy and Security- Research Paper Series* Research paper(4):411–419.

Trottier, D. 2014. Crowdsourcing cctv surveillance on the internet. *Information, Communication & Society* 17(5):609–626.

Yin, M., and Chen, Y. 2015. Bonus or not? learn to reward in crowdsourcing. In *Proc. of IJCAI*, 201–207.

Yin, M.; Chen, Y.; and Sun, Y.-A. 2014. Monetary interventions in crowdsourcing task switching. In *Proc. of HCOMP*.

Zeelenberg, M., and Beattie, J. 1997. Consequences of regret aversion 2: Additional evidence for effects of feedback on decision making. *Organizational Behavior and Human Decision Processes* 72(1):63–78.