Optimizing Interventions via Offline Policy Evaluation: Studies in Citizen Science

Avi Segal, Kobi Gal
Ben-Gurion University of the Negev, Israel

Ece Kamar, Eric Horvitz
Microsoft Research, Redmond WA

Grant Miller
University of Oxford, U.K.

Abstract
Volunteers who help with online crowdsourcing such as citizen science tasks typically make only a few contributions before exiting. We propose a computational approach for increasing users' engagement in such settings that is based on optimizing policies for displaying motivational messages to users. The approach, which we refer to as Trajectory Corrected Intervention (TCI), reasons about the tradeoff between the long-term influence of engagement messages on participants’ contributions and the potential risk of disrupting their current work. We combine model-based reinforcement learning with offline policy evaluation to generate intervention policies, without relying on a fixed representation of the domain. TCI works iteratively to learn the best representation from a set of random intervention trials and to generate candidate intervention policies. It is able to refine selected policies off-line by exploiting the fact that users can only be interrupted once per session. We implemented TCI in the wild with Galaxy Zoo, one of the largest citizen science platforms on the web. We found that TCI was able to outperform the state-of-the-art intervention policy for this domain, and significantly increased the contributions of thousands of users. This work demonstrates the benefit of combining traditional AI planning with off-line policy methods to generate intelligent intervention strategies.

Introduction
Volunteer-based crowdsourcing has been harnessed to engage thousands of people in solving challenges online. Examples include citizen science applications like Foldit (Khatib et al. 2011), e-bird (Sullivan et al. 2009) and Zooniverse (Simpson, Page, and De Roure 2014), as well as question and answer sites like stack overflow (Anderson et al. 2012). A large majority of people coming to these sites only make a few contributions before leaving (Preece and Shneiderman 2009; Varshney 2012). We address the challenge of engagement in such systems through adaptive interventions, aimed at unlocking additional value that would come with more sustained contributions (Eveleigh et al. 2014; Segal et al. 2015). We show the value of generating interventional policies based on joining model-based reinforcement learning with offline policy evaluation.

We formalize the task of computing effective interventional policies as a problem in sequential decision making under uncertainty, where an agent can choose whether to generate one of several possible motivational messages to users at a given point in time. Interventions are associated with a cost of interruption that can interfere with the user’s work (Horvitz, Jacobs, and Hovel 1999). Thus, the agent needs to manage the tradeoff between intervening at a current state versus waiting to collect more information and taking the risk that the user will disengage from the system. The agent also needs to balance the long-term benefits and short-term disruptions associated with different intervention actions.

The online nature and quick turnaround of individual users in volunteer-based crowdsourcing poses new challenges for optimizing intervention decisions. We do not know the dynamics governing people’s online behavior and their responses to potential interventions. However, efforts to learn a good policy online by performing experiments via interventions may disrupt the work of volunteers and contribute to early disengagement.

We address these challenges by applying a combination of techniques from model-based reinforcement learning and offline policy evaluation on historical data collected previously from trials with random interventions. We search iteratively for a representation that succinctly maps histories to states. We build on previous efforts that have used offline policy evaluation to compute non-biased estimates of the value of a given policy using an existing set of random trials (Precup 2000; Mandel et al. 2014). We show with experiments that traditional uses of importance sampling can be arbitrarily noisy when applied to human interaction data.

We extend these approaches by providing an offline methodology for correcting candidate policies, under the constraint that users can be interrupted only once during a session in order to bound the potential disruption.

Our approach, called Trajectory Corrected Intervention (TCI), searches iteratively for the representation that leads to the best intervention policy. For any candidate representation, TCI builds a corresponding MDP based on a training set taken from past trajectories and solves the MDP to extract a target policy. The resulting target policy is evaluated using importance sampling on a validation set taken from the past trajectories. The search terminates when perturbing
the representation does not yield further improvements to the expected value of the policy. The resulting policy is subsequently evaluated in a policy-correction step for each state, on a test set taken from the past trajectories. The correction step exploits the structure of the domain, in which only a single interruption is possible in each session (a limitation dictated by the community leaders), to statistically validate if the action selected by the target policy indeed provides better value than alternatives. This procedure replaces an intervention action with an alternative intervention action (or a decision not to intervene) when the alternative action yields higher value on the trajectory history.

We implemented the TCI approach on Galaxy Zoo, one of the largest citizen science platforms in the world, where volunteers are asked to classify celestial bodies drawn from the massive Sloan Digital Sky Survey (SDSS). Analyses of Galaxy Zoo logs have shown that the vast majority of users leave quickly and make only a few contributions. We examine the value of providing personalized motivational messages aimed at increasing the contributions of users. We consider how best to balance intervening immediately with a motivational message, based on the current state of information about the participant, with waiting to collect additional information and risking the loss of the user.

TCI learned a representation that includes features that summarize users’ behavior in the domain, as well as a belief state that measures the probability that the user will disengage from the system. The TCI approach identified a policy of choosing either one of three motivational messages or no intervention at each state. Our experiments, which were performed in the wild on the Galaxy Zoo platform, showed that TCI was able to outperform an earlier myopic approach, by considering the long term effects of the intervention messages. We also found that the policy correction step is critical; the corrected policy achieved significant gains in user productivity when deployed in the live system compared to the target policy generated with a version of TCI without the correction step.

We make three key contributions: First, we provide an end-to-end method for computing optimal intervention policies with application to volunteer-based crowdsourcing. The policies are based on an analysis of past trajectories, and do not rely on a specific representation. Second, we provide a new correction method that can address the errors associated with applying offline policy evaluation to Galaxy Zoo by exploiting the structure of the domain. Third, we show the real-world influence of the methods, by significantly extending the engagement and contributions made by thousands of volunteers in the Galaxy Zoo platform.

### Related Work

Our approach builds on prior work in two separate fields of research: modeling and extending engagement in crowdsourcing and off-line policy evaluation in reinforcement learning.

There is a growing interest in methods for motivating users in volunteer based crowdsourcing (Evlelegh et al. 2014; Jackson et al. 2014). We consider several studies of computational approaches for describing and extending user engagement in online communities. Anderson et al. (2013) used badges to steer behavior towards required goals in question-answer sites. They developed a model of behavioral change that is induced by badges for the stackoverflow site. Their model showed that change in user behavior increases as the badge frontier gets closer, and was able to predict observations about the real-world behavior of user on stackoverflow. In subsequent work, Anderson et. al (2014) performed a large-scale deployment of badges as incentives for engagement in a MOOC, including randomized experiments in which the presentation of badges was varied across subpopulations.

Mao et al. (2013) developed a predictor of the disengagement of participants in Galaxy Zoo. Their study considered different features including statistics about volunteers’ characteristics, the tasks they solved, and their history of prior sessions on the system. They demonstrated the effects of different session lengths and window sizes on the accuracy of the predictions about the timing of disengagement.

Segal et al. (2016) studied three different intervention messages on the volunteers of Galaxy Zoo when the messages were timed according to predictions of their disengagement. A controlled study showed that the combination of a motivational message emphasizing the individual contribution of users and its prediction-based timing was able to generate the highest engagement levels from users, when compared to alternative messages that emphasized users’ sense of community and relieved their anxiety about making mistakes. The work presented here builds on this line of work and shows that the TCI approach was able to achieve significantly better results than this myopic method, by optimizing the intervention policy over all message types and long term effects.

Other relevant efforts come from the literature on interruption management and retention modeling. Horvitz et al. (1999) present a decision-theoretic approach to balancing the cost of interruptions with the cost of delay in the transmittal of notifications. Horvitz and Apacible (2003) used machine learning to infer the cost of interrupting users over time given data from their online interactions, calendars and visual and acoustical analyses. Shrot et al. (2009; 2014) used collaborative filtering to predict the cost of interruption by exploiting the similarities between users and used this model to guide an interruption management algorithm. Rosenfeld and Kraus (2016) motivated and persuaded users in argumentative dialog settings using a POMDP based model and machine learning based predictions. Azaria et al. (2014) considered the problem of automatic reward determination for optimizing crowd system goals and presented two algorithms that outperformed strategies developed by human experts.

In offline policy evaluation, a target policy is evaluated using a pre-collected dataset that was generated via execution of a different behavioral policy (Thomas, Theocharous, and Ghavamzadeh 2015; Thomas and Brunskill 2016; Liu, Mahadevan, and Liu 2012; Peshkin and Shelton 2002). This approach is common in many settings involving human interactions where it is not possible to probe users online (e.g., patient diagnosis systems and e-learning). Many approaches
for solving the off-line policy evaluation problem have used sampling techniques to compute the value of target policies. Precup (2000) introduced several importance sampling estimators for the value of a target policy, by weighing samples according to the ratio of the likelihood between the target policy and the behavior policy. Jiang et al. (2015) extended a bandits’ approach to estimating values of the target policy. They produced non-biased estimators of the true policy value that may exhibit lower variance than using traditional importance sampling techniques.

Most relevant to our approach is the work by Mandel et al. (2014; 2016) who used sampling methods for off-line policy evaluation across several candidate representations in educational games. We extend this approach in two ways. First, we provide a search based optimization across representations, focusing on binary representations for continuous features. Second, we introduce a correction mechanism to decrease variance in our generated policies and show its efficacy in a large scale deployment.

Lastly, we consider the optimal stopping literature in statistics, which studies the problem of timing a termination action in order to maximize an expected reward. Our intervention problem has the special structure that a single intervention is possible at a given session. Thus the problem can alternatively be formalized as an optimal stopping problem where the reward for taking an intervention action is uncertain and can be estimated based on trajectories. Tractable algorithms (e.g., threshold-based methods) exist for classes of optimal stopping problems where the world dynamics have a known, well-characterized structure (Peskir and Shiryaev 2006). However, these tractable algorithms are not applicable to human-interaction settings where transitions have an arbitrary form. These problems can then be solved through existing MDP solutions techniques such as dynamic programming (Monahan 1982), which we carry out in this work.

**Problem Description and Approach**

We consider a setting with two actors: a user who is repeatedly interacting with a system to complete tasks and who can disengage at any time at will; and an agent that can intervene in real time, presenting messages to the user with the goal of increasing her contributions. We start by providing definitions that are used in our formalization. A user episode (session) of length $T$ consists of a sequence of agent actions, observations and rewards. At each timestep $t \in \{1, \ldots, T\}$ the agent performs an agent action $a_t$ which consists of one of several possible intervention actions (e.g., generating a motivational message in Galaxy Zoo) or a no-op action (no intervention). The user generates an observation $o_t$ (e.g., classifying a galaxy), and the agent incurs a scalar reward $r_t$ (e.g., the quality of the classification). The history at timestep $t$ is denoted $\{(o_1, r_1, o_1), \ldots, (o_t, r_t, o_t)\}$. An agent can interrupt the user at most once per episode. There exists at most a single timestep $i$ in which $a_i$ is an intervention action, which consequently influences the rewards and observations in the future time steps.

---

**Algorithm 1: The TCI Approach**

<table>
<thead>
<tr>
<th>Data: Domain description $B$, feature set $F$, past trajectories $D = D_{train} \cup D_{val} \cup D_{test}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result: Optimized representation $M$, Target Policy $\pi_M^*$</td>
</tr>
<tr>
<td>1 $EV^* \leftarrow 0$</td>
</tr>
<tr>
<td>2 repeat</td>
</tr>
<tr>
<td>3 $M \leftarrow \text{GetNextRepresentation}(B, F)$</td>
</tr>
<tr>
<td>4 $\pi_M \leftarrow \arg\max_{\pi} EV[\pi, M, D_{train}]$</td>
</tr>
<tr>
<td>5 $EV(\pi_M) \leftarrow \text{ComputeVal}(\pi_M, D_{val})$</td>
</tr>
<tr>
<td>6 if $(EV(\pi_M) &gt; EV^*)$ then</td>
</tr>
<tr>
<td>7 $EV^* \leftarrow EV(\pi_M)$</td>
</tr>
<tr>
<td>8 $\pi_M^* \leftarrow \pi_M$</td>
</tr>
<tr>
<td>9 until convergence;</td>
</tr>
<tr>
<td>10 $\pi_M^* \leftarrow \text{CorrectPolicy}(B, \pi_M^*, D_{test})$</td>
</tr>
<tr>
<td>11 return $M$, Policy $\pi_M^*$</td>
</tr>
</tbody>
</table>

---

The overall TCI approach is outlined in Algorithm 1. The input to the TCI process is (1) a domain description $B$ that includes a set of agent actions, user observations and rewards; (2) a set of features $F$ that are aggregations over histories, and used to create the state space; (3) a dataset $D$ of past trajectories that are composed of histories of random agent actions and their observed effect on user behavior in the system. The policy that generated these trajectories is called the "behavioral policy". This data is divided to separate training, validation and testing sets.

The TCI approach consists of three main steps, which we outline below. Step 1 (lines 3-4) integrates two optimization tasks: finding the optimal representation for the intervention domain, and extracting the best policy given this representation. A representation $M$ is a many-to-one mapping from histories of interactions to states. When $M$ includes the full history, it provides a complete description of the domain, but the size of the representation makes the data too sparse to learn from. Instead, $M$ provides a reduction of the state space to ranges over subsets of the features in $F$. We detail this step in the next section. We learn an MDP over the representation $M$ given the training set and extract the current target policy $\pi_M$ (line 4).

Step 2 (line 5) estimates the value of the target policy $\pi_M$ on the validation set. We iteratively execute Steps 1 and 2 to find the next representation that improves the value of the extracted target policy. The process terminates when successive steps fail to improve the value of the policy for a designated number of iterations. Step 3 (line 10) corrects the policy $\pi_M$ for errors by comparing its performance to choosing alternative intervention decisions (or a decision not to intervene) at each state. The output of the TCI process is the optimized representation $M$ and its associated target policy $\pi_M^*$.

**Implementation: Galaxy Zoo**

We now describe how we have applied the TCI approach to Galaxy Zoo. A user session in Galaxy Zoo includes an episode with discrete timesteps from 0 (logging on) and $T$.
Thousands of people are taking part in the project every month. Visit Talk at talk.galaxyzoo.org to discuss the images you see with them.

We use statistical techniques to get the most informative features from the study of Segal et al. (2016). This data is divided into training, validation and test sets as summarized in Table 2. In generating random interventions, the timestep of the agent actions also include a fourth action which is a no-op (inactivity). At each discrete time step \( t \leq T \) the agent chooses an action (whether to generate a motivational message at this timestep, and if so which message). The reward \( r_t \) at time \( t \) is 1 when a user classifies a galaxy and 0 otherwise. The observation at each timestep included 16 features over the user’s history and current session (Mao, Kamar, and Horvitz 2013). The most informative features were: the number of session counts for the user in the system, the number of completed tasks in the current session, the number of completed tasks averaged over all sessions, the number of seconds spent in the current session, the number of seconds per session averaged over all sessions, the average dwell time in the current session (i.e., the average number of seconds between two consecutive task submissions by the user).

An additional observation is the probability that the user will disengage within a 5-minute time window, computed using these features. This predictor serves as a proxy for the motivation of the user.

The set of intervention actions for the agent includes three motivational messages displayed in Table 1. These messages directly address motivational issues affecting users’ performance in citizen science (Eveleigh et al. 2014). The “helpful” type message emphasizes users’ contribution to the project, the “community” type message emphasizes the collective nature of the project, and the “anxiety” type message emphasizes the tolerance to individual mistakes. The agent actions also include a fourth action which is a no-op action, deciding not to intervene at the current state.

The trajectory history consists of an expanded version of the dataset of randomized intervention trials collected from the study of Segal et al. (2016). This data is divided into training, validation and test sets as summarized in Table 2. In generating random interventions, the timestep of the motivational messages was drawn uniformly between the limits of 0 (i.e., intervene immediately) and a session length that was sampled from a Poisson distribution that was fit to historical Galaxy Zoo participation rates. Data and accompanying information to this paper can be found at http://tinyurl.com/ztujcvz.

### Table 1: Intervention messages used in the study.

<table>
<thead>
<tr>
<th>Type</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Helpful</td>
<td>Please don’t stop just yet. You’ve been extremely helpful so far. Your votes are really helping us to understand deep mysteries about galaxies.</td>
</tr>
<tr>
<td>Community</td>
<td>Thousands of people are taking part in the project every month. Visit Talk at talk.galaxyzoo.org to discuss the images you see with them.</td>
</tr>
<tr>
<td>Anxiety</td>
<td>We use statistical techniques to get the most from every answer. So, you don’t need to worry about being “right”. Just tell us what you see.</td>
</tr>
</tbody>
</table>

### Table 2: Dataset of randomized intervention trials.

<table>
<thead>
<tr>
<th></th>
<th>Users</th>
<th>Interventions</th>
<th>Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>2,302</td>
<td>3,265</td>
<td>245,695</td>
</tr>
<tr>
<td>Validation</td>
<td>1,722</td>
<td>1,730</td>
<td>114,788</td>
</tr>
<tr>
<td>Test</td>
<td>1,281</td>
<td>2,173</td>
<td>119,457</td>
</tr>
</tbody>
</table>

The initial set of features used in the TCI process included the six prominent features mentioned in the previous section, as well as the predicted probability that the user will disengage (which used the entire set of 16 features). We hypothesized that the TCI approach would learn a succinct representation of the domains using a subset of these features while still providing an intervention policy with high value.

The representation \( M \) induces an MDP over the state space. To learn the MDP parameters, we use the training set of the trajectory history. The transition function \( T \) of the MDP is computed as the expectation over the observed transitions in the training set given representation \( M \).

If \( a \) is an intervention action (not a no-op), then the system transitions to the terminal state with probability 1. The reward for an action at a given state \( s_t \) depends on whether the action is an intervention action or a no-op, and whether \( s_t \) is a terminal state.

We now describe how to compute the reward. Let \( m \) be all of the episodes that match the state-action pair \((s_t, a)\). The reward associated with an intervention action \( a \) at time \( t \) is

\[
R(s_t, a) = 1 + \frac{1}{m} \sum_{i=1}^{m} R_i
\]

where \( R_i = \sum_{k=t_i+1}^{T} \delta^{k-t_i-1} \cdot r_k \). Here \( t_i \) is the timestep in episode \( i \) where intervention action \( a \) was given in step \( s_t \), \( \delta \) is the discount factor, and \( r_k \) is the reward at episode \( i \) at time \( k \). If \( a \) is a no-op action then we assign \( R(s_t, a) = 1 \) (user performed one contribution in this state). Lastly, transitioning to the terminal state with a no-op action represents a user disengaging from the system and is assigned a reward of zero.

We solve the MDP to compute a target policy \( \pi_M \) for the given representation using value iteration.

To find the optimal policy representation, we perform search optimization over the representation cutoff values \( \{v_1, \ldots, v_n\} \) using the Particle Swarm Optimization (PSO) algorithm (Polli, Kennedy, and Blackwell 2007). PSO is an evolutionary algorithm for optimizing a problem’s solution by iteratively searching over a candidate space with regard to a given measure of quality (in our case the value of a policy). We used parameter values recommended by Pedersen et al. (2010) with a swarm size of 100 particles and a maximum of 40,000 fitness evaluation steps. Stopping was performed if the evaluation steps limit was reached or if fitness did not improve in the last 100 iterations (set empirically).

When run on the training set \( D_{train} \) of past trajectories, the TCI approach converged on a representation that included the following four features related to user activities in a current session: The number of tasks \( (s_{sessionTasks}) \), the number of active seconds in this session \( (s_{sessionTime}) \), the dwell time \( (s_{avgDwell}) \) and
the disengagement prediction \((s, \text{dis\_pred})\). Figure 1 shows a sample from the extracted best policy for users who engaged in a single session with the system (about 70% of the user population). For example, in state 5 (corresponding to the state where the user is productive, works slowly, and is not likely to leave) the system generated the community based message (“Thousands of people are taking part in the project...”).

**Step 2: Computing the Value of a Policy**

Computing the value of a target policy on the validation set (line 5 in Algorithm 1) is an instance of the off-policy evaluation problem (Thomas, Theocharous, and Ghavamzadeh 2015). Specifically, the distribution over states that is induced by the target policy \(\pi_M\) is different than the distribution over states in the randomized training set, induced by the “behavioral” policy (the policy used to create the dataset at hand). A common approach is to use sampling techniques to correct for this discrepancy by assigning higher weights to samples from the target policy that differ from the behavioral policy (Precup 2000). A main advantage of the importance sampling techniques is that it is consistent, i.e., it provides a non-biased estimate of the true value of the policy.

The input to the importance sampling step is a behavior policy \(\pi_b\), a dataset of trajectories \(D_{\text{valid}}\) and a target policy \(\pi_M\) we want to evaluate. The output is the estimated value of the target policy on the validation data set described in Table 2.

Let \(\pi_M\) be a target policy, and \(H\) be a history of \(m\) episodes. To apply importance sampling in TCI, we need to define how we compute the likelihood of a target policy on an episode. For any policy \(\pi\) let \(P_\pi\) be the induced probability distribution assigned by policy \(\pi\) over all agent actions in action set \(A\). The likelihood of a policy \(\pi\) for a history \(h_j \in H\) at timestep \(t\) is defined as

\[
P_\pi(a_1, \ldots, a_t \mid h_t) = \prod_{j=1}^{t} P_\pi(a_j \mid s_j) \tag{2}
\]

where \(s_j = M(h_j)\) is the state that corresponds to history \(h_j\) according to representation \(M\).

We use the approach of Precup (2000) and Mandel (2014) to compute the expected value of the target policy for episodes of increasing lengths. We assign higher weights to samples that are less likely according to the behavioral policy but more likely according to the target policy. The expected value of a target policy \(\pi\) on a history \(H\) of \(m\) episodes of maximal length \(T\) given behavioral policy \(\pi_b\) and representation \(M\) is

\[
EV(\pi_M \mid H) = \sum_{t=1}^{T} \sum_{i=1}^{m} \sum_{j=1}^{m} P_{\pi_M}(a_{i,1}, \ldots, a_{i,t} \mid h_{i,t}) \cdot \delta^{t-1} r_{i,t} \tag{3}
\]

Here, \(a_{i,t}\) and \(r_{i,t}\) refer to the agent action and reward taken at episode \(i\) at time \(t\), respectively. In our study, the behavioral policy \(\pi_b\) is stochastic: it assigns a probability distribution over agent actions for each possible state, whereas the target policy \(\pi_M\) is deterministic, it assigns a probability of 1 or 0 to a given action and state pair.

**Step 3: Policy Correction**

The unbiased nature of offline policy evaluation based on importance sampling does not guarantee that value estimates are correct. Due to the sparsity in historical data, value estimates for a given policy can be noisy (Jiang and Li 2015). To analyze the behavior of importance sampling in the domain, we generated a simulator based on data collected from Galaxy Zoo.

The simulator learned distributions representing user activity and user response to interventions using the random intervention dataset. The goal for creating the simulator was to have an experimental domain where we could compute the ground truth value of any policy, and thus observe the errors in the value estimates of offline policy evaluation.

We computed the absolute error between the importance sampling estimator of the value of a target policy, and the actual value of the policy once executed in the simulation. Figure 2 plots the error for different representations \((y\text{-axis})\) given the trajectory support, which is the likelihood similarity between a target policy and behavioral policy used to generate the simulation \((x\text{-axis})\). We observe high variance in the errors generated by the importance sampling estimator. We also note that the lower error values are not necessarily for the highest supported trajectories and that there are high value estimates (darker points) which have high support and suffer from high error. This means that our offline optimization process may yield policies that may not be optimal when applied to Galaxy Zoo in real time.

A non-optimal target policy will include suboptimal actions in some states. Namely, for state \(s_t\) and action \(\pi_M(s_t)\), there exists an agent action \(a \neq \pi_M(s_t)\) which is “better”
is associated with the optimal action for each state. Given that we only interrupt once per session, the value that intervention actions at that state using statistics from data. assigned by the target policy for a given state to alternative
tise, and consistency across experts in not guaranteed.

An alternative approach is to compare the value of actions
H
A
\pi
\pi
A
\pi
H
\pi
A
EV
\pi
H
EV
[st, no-op* | H] is a lower bound for choosing no-op in st and possibly intervening in the future. Thus, if EV[st, no-op* | H] > EV[st, \pi_M(st) | H], then this means EV[st, no-op | H] > EV[st, \pi_M(st) | H], and we can replace \pi_M(st) with no-op. We thus consider an alternative set \hat{A} of agent actions that include all intervention actions as well as the special no-op* action. Note that when the no-op* bound is loose, we may fail to correct a policy when the utility of the current action is lower than the utility of no-op but is higher than no-op*.

This process is described in Algorithm 2. The input to the correction process is a target policy \pi_M, the test set trajectories of Table 2 and the set of agent actions \hat{A}. The output of this process is a corrected policy which may replace intervention actions in given states with other intervention actions or with a no-op action. We implemented the correction

Algorithm 2: Policy Correction for Intervention Policy

Data: Target policy \pi_M, trajectories D_{test}, intervention actions \hat{A}.

Result: Corrected deterministic target policy \pi^+_M

1 forall \ s_t \in \ S do
2 \ \pi^+_M(s_t) \leftarrow \text{argmax}_{a \in \hat{A}} \text{EV}((s_t, a) | D_{test})
3 return \pi^+_M

approach on the target policy computed in Steps 1 and 2 of Algorithm 1. Figure 3 shows a sample from the corrected policy for first session users. We highlight the changed actions (in red) proposed by this correction step. For example, in state 5, the system corrected the intervention from the community based message (“Thousands of people are taking part in the project every month...”) to the anxiety based one (“you don’t need to worry about being right”).

Empirical Studies
We conducted two separate studies to evaluate the effect of the TCI approach. Both studies were based on interventions that were performed in real time in the Galaxy Zoo domain. In all studies, the discount factor \delta was set to 0.95 (determined empirically). The running time of computing the TCI optimized policy for the dataset in Table 2 on a Mac
Influence of Optimized Approach

In the first experiment, we compared the effects of our optimized approach to alternative intervention policies, including the approach of Segal et al. (2016) which represents the state of the art. The study was run between May 8 and June 21, 2017 and included a total of 3,383 users. All users logging on to the system during this period of time were randomly divided among the following cohorts: (1) Users receiving messages according to the TCI optimal intervention strategy (Optimized-Policy-Corrected Group); (2) Users receiving a helpful intervention message when they are predicted to disengage (Myopic-Policy Group). This is the policy suggested by Segal et al. (2016) (3) Users receiving intervention messages according to a random policy (Random-Intervention Group). (4) Users receiving no intervention (Control Group). Each of the cohorts included 864 participants, except the Myopic group which included 865 participants. In total, 3,755 interventions were generated for all of the intervention cohorts. We computed the expected number of interventions for each condition and ensured that the number of generated interventions for each cohort was the same.

Figure 4 shows the distribution over the intervention actions generated by the TCI policy. As shown by the figure, the most common message-type generated by the TCI approach was the helpful message (56%), followed by the community message type (32%) and anxiety message type (12%).

A natural question to ask is whether optimizing intervention decisions based on the TCI approach was beneficial. Figure 5 compares the average contribution rates for users in the three cohorts. We required that (1) for all cohorts, users received at least one intervention message, and (2) in the optimized-policy corrected cohort, users received at least one intervention message of the anxiety- or community-type message (different than the helpful-type message used by the Segal et al. (2016) baseline). As can be seen in the figure, the users in the Optimized-Policy-Corrected group generated 69% more contributions than users in the Myopic-Policy group ($p < 0.05$, ANOVA). Users in both of these groups made more contributions than those in the random-intervention group and in the control group (the control group performance was not significantly different than that of the random group and is not shown in the figure).

A potential explanation of the additional influence of the community and anxiety messages is that they resonate with participants’ needs and fears at the right time during their engagement with the system (Segal et al. 2015). Nonetheless, without controlling the timing of the intervention based on predictions of forthcoming disengagement and additional factors, these messages are not effective, as demonstrated by the Random condition.

Effect of Correction Step

We now report on a study of the effect of the correction step, in isolation, on the performance of our approach. To this end we conducted a separate experiment for comparing between the target policy obtained in Step 2 of the TCI process with the corrected policy obtained in the final Step 3. The study was run between June 22 and August 10, 2017. Users logging on to the system during this time period were randomly divided between the two cohorts: Users receiving the TCI intervention policy after policy correction (916 users) and before correction (917 users).

Figure 6 shows the contribution rates of users with 1 session (the majority of users and where the policy correction step performed most of the corrections) which received an intervention for the different cohorts. As shown in the figure, the average contribution rate for users in the Optimized-Policy-Corrected group was significantly higher than that of users in the Optimized-Policy-Uncorrected group ($p < 0.05$, t-test). This result demonstrates the crucial effect of the correction step on users’ contribution rates.

Discussion and Conclusion

We have provided a new computational method called TCI for increasing engagement in volunteer based crowdsourcing. The input to the TCI approach includes a domain description and a set of history trajectories of random intervention trials. TCI iteratively searches for the optimal intervention policy for the domain by combining model-based reinforcement learning with off-line policy methods. The pol-
ic policy is then corrected, leveraging the constraint of allowing only a single interruption per session. The need to minimize costs of interventions and of the use of single interventions per session extends to other domains (e.g., e-learning, mobile health). We tested our approach in a live experiment in the Galaxy Zoo domain, a large-scale citizen science platform where users classify celestial galaxies. We demonstrated that our approach significantly outperforms a state-of-the-art baseline.

We mention several limitations to our approach and subsequent suggestions for future work. First, TCI relies on a set of random intervention trials for training the MDP and evaluating and correcting candidate policies off-line. In many time critical domains (e.g., citizen science, healthcare), the cost of performing random intervention trials may be unacceptable. Other approaches for providing trajectory histories can use simulations or domain experts. The data collection step can also leverage active research on finding the minimal number of random interventions required to reach statistically significant effects in intervention design for healthcare applications (Klasnja et al. 2015). We hope to see similar models developed for crowdsourcing domains. Second, we noted that there were lower average contribution rates in the correction step experiment compared to the first experiment. We attribute this to the summer period in the northern hemisphere, which highlights challenges around changing domain dynamics. The dynamics of participation and engagement in the Galaxy Zoo domain (e.g., changes in contribution distributions across the year) makes it an interesting experiment platform for future studies. Third, the TCI approach has assumed that a single interruption is allowed per session. Allowing more than a single interruption will require to adapt step 1 (representation and optimization) and step 3 (policy correction). Specifically, the state space will be augmented to include the number of interruptions generated for the user and the MDP transition matrix will be updated accordingly. Additionally, the correction step will need to be reformulated to account for the fact that multiple interruptions are allowed. Finally, while the TCI approach had a significant positive influence on the behavior of thousands of users in Galaxy Zoo, it still needs to be extended and tested in other domains. We are working on such an extension to the e-learning domain.

We are excited about opportunities to leverage offline policy optimization to enhance engagement in citizen science and other volunteer-centric online applications. Beyond these applications, the methods can be valuable in other kinds of engagement challenges, such as in educational systems, where interventions, for both motivation and for assisting with inferred conceptual challenges, may enhance learning experiences and efficacies. Before concluding, we note the need to be vigilant about potential societal challenges rising with uses of methods that seek to optimize engagement of people when it comes to goals of financial or political gain.

Acknowledgments

This work was supported in part by the SOCIAM grant, funded by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/J017728/1. We thank Alex Bowyer for building the intervention API for Galaxy Zoo.

References


