

# Harvard University, John A. Paulson School of Engineering and Applied Sciences

## CS236r Final Project Computational Results For Peer-Prediction and Crowdsourced Judgement Elicitation

Virgile Audi (vaudi@g.harvard.edu)  
Charles Liu (cliu02@g.harvard.edu)

May 4, 2016

### Introduction

A common theme throughout the course is to question the practicality of mechanisms discussed in class. For our presentation of *Crowdsourced Judgement Elicitation with Endogenous Proficiency* and *Eliciting Informative Feedback: The Peer-Prediction Method*, we found the mechanisms relied on strong assumptions that were generally unrealistic. Also, as seen in the second homework with specifying learning rates, in many situations a model can be very sensitive to its parameterization. Difficulties in finding proper parameters can also be viewed as a potential flaw of the mechanism, making it unsuitable for use in a practical setting. For our final project we wanted to take a deeper look at the two mechanisms, relax their assumptions, and see if the mechanism can be modified to achieve similar results. All work can be found at <https://github.com/chuckyoliu/Crowdsourced-Elicitation>.

### 1 Crowdsourced Elicitation

The *Crowdsourced Judgement Elicitation with Endogenous Proficiency* paper dealt with a model that incorporated three variables: effort, proficiency, and truthfulness. A user would be given a set of tasks, with each task having a binary type H or T. A user first has the binary option of giving full or no effort. On no effort, the user reports H/T based on a random coin flip. On full effort, the user then has a probability corresponding to his proficiency of correctly interpreting the event. Finally, he can report his signal either truthfully or untruthfully.

An example use case for this kind of mechanism would be in grading assignments for massive open online courses (MOOCs). If there was a need for graders, e.g. assignments with non-numerical or multiple choice answers, it would be infeasible for a staff to consistently grade answers from thousands of students. Instead, having peer grading would be a natural solution conditional on the fact that students will take the time to grade the assignments and truthfully report the scores.

#### 1.1 Computational Results of Paper

We first confirmed the results from the paper computationally, given all of their assumptions. The main benefit to this mechanism was that the maximum reward was achieved in the equilibrium where everyone gave full effort and reported truthfully, whereas giving no effort resulted in an expected reward of 0. Giving full effort and reporting untruthfully is also an equilibrium, however this can be remedied by having some “known” truthful users put into the mechanism. One of the stronger results from the paper was Lemma 8:

Suppose the probability of agent  $i$  using strategy  $(1, X)$  is  $\delta$  and strategy  $(0, r_i)$  is  $1 - \delta$  for each task  $j \in J(i)$ . Suppose  $i$ 's potential reference raters  $r_j(i)$  use strategies  $(1, X)$  and  $(0, r_{r_j}(i))$  with probabilities  $\epsilon_{r_j}(i)$  and  $1 - \epsilon_{r_j}(i)$  respectively, for each task  $j \in J(i)$ . If  $\epsilon_{r_j}(i) > 0$  for any reference rater with proficiency  $p_{r_j}(i) > \frac{1}{2}$ , then agent  $i$  has a (strict) profitable deviation to  $\delta' = 1$ , i.e., to always using strategy  $(1, X)$ , for all values of  $r_i \in [0, 1]$

Figure 1: Computational results of different equilibriums vs. other strategies

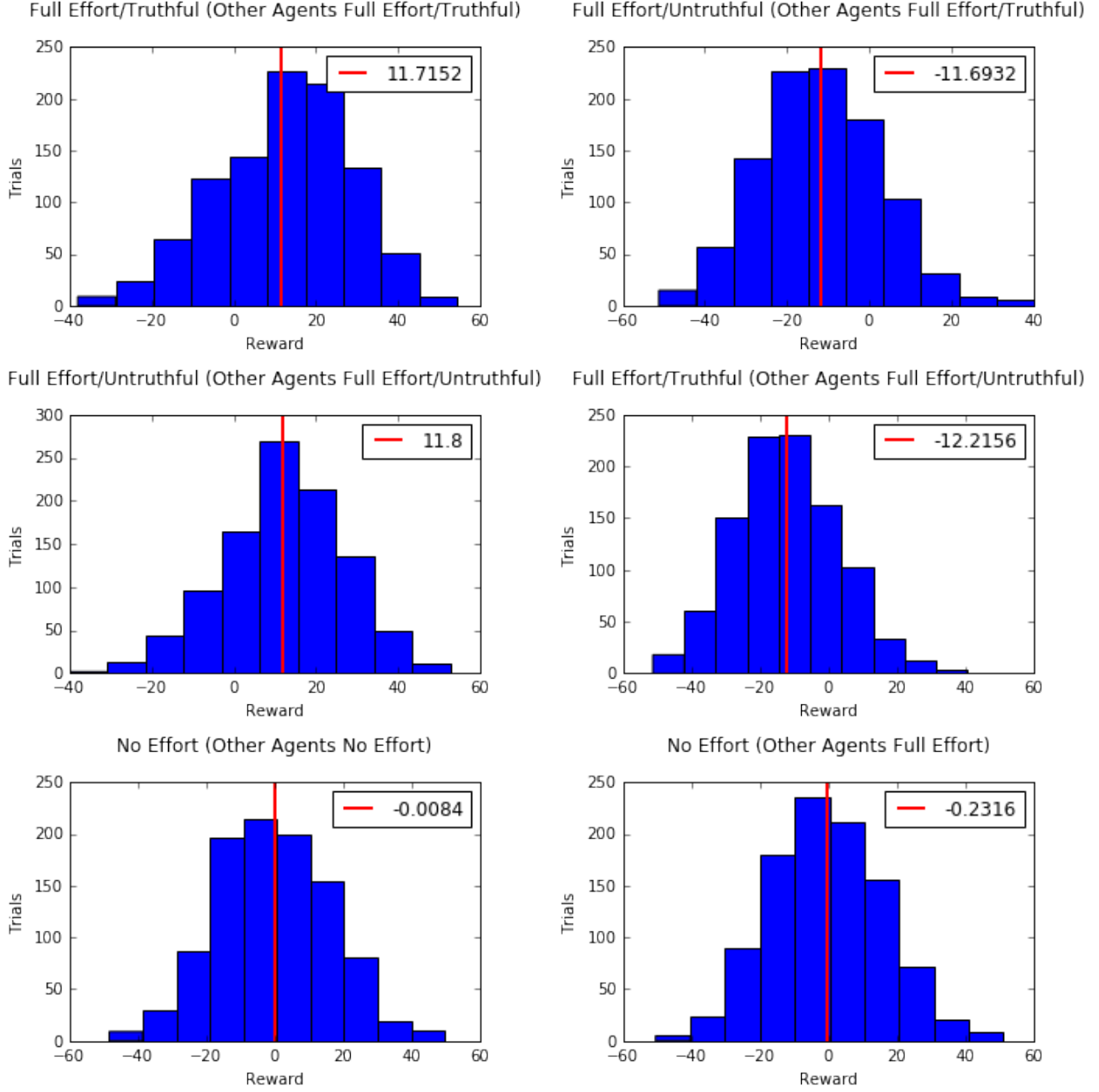


Figure 2: Computational results of Lemma 8

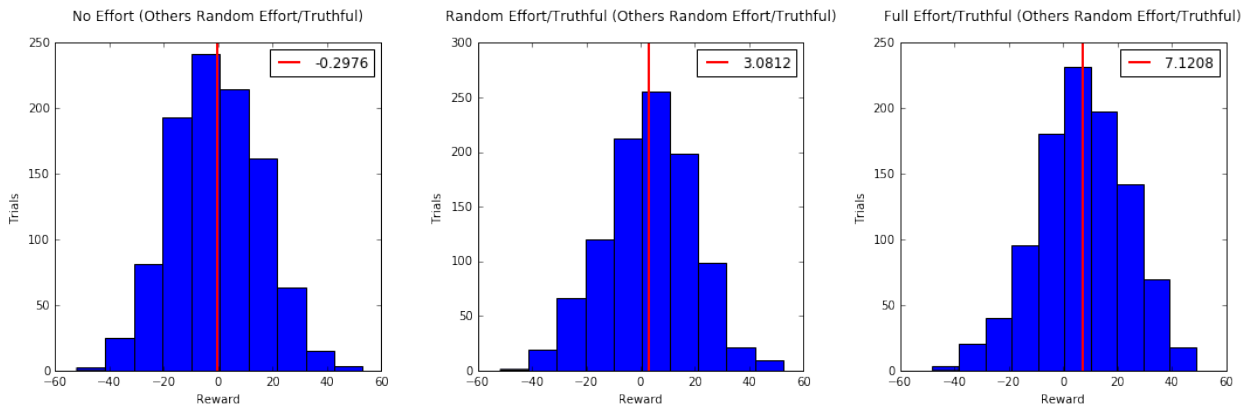


Figure 1 is a summary of the main results and Figure 2 of Lemma 8. In each scenario, there were 100 tasks and 20 agents each performing 10 tasks. Simulations were run 1000 times, rewards were scaled by 10, and no costs were factored into the model. All agent proficiencies were randomly assigned between 0.5 and 1.

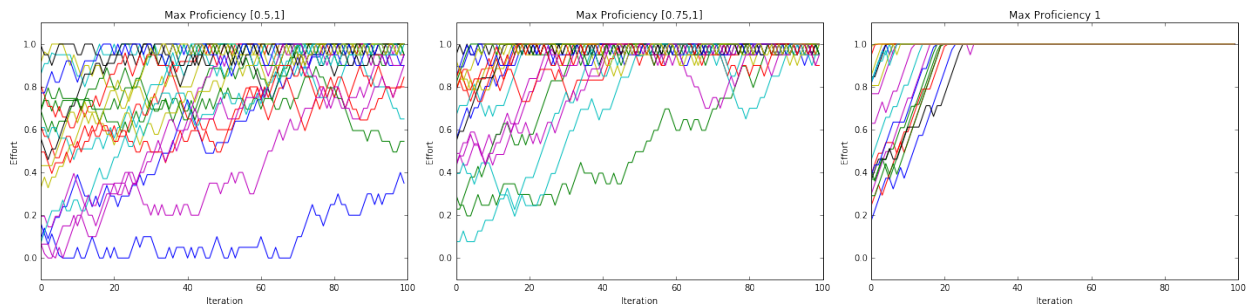
## 1.2 Continuous Effort

The paper also noted that all results could be extended to the case of a continuous effort distribution where proficiency increased linearly. For a given agent with max proficiency  $p_m$ , proficiency for a given effort level  $e$  was calculated as:

$$prof(e) = 0.5 + e * (p_m - 0.5)$$

This maintains the properties of the binary effort case: when effort is fully given your proficiency corresponds to the max proficiency, and when no effort is given your proficiency is in effect a random coin flip. To test that giving full effort was still an equilibrium, we randomly assigned effort levels to each agent. Then iteratively, we would select one agent and perturb its effort level up and down by 0.05 and move the agent’s effort level to whatever returned the highest reward. No costs were factored in and rewards were scaled by a factor of 10, similar to the original results.

Figure 3: Computational results from having continuous effort



An equilibrium at full effort is generally reached, but the convergence is not immediate for agents with a lower max proficiency. Figure 3 displays the trace of efforts for each agent under three scenarios:

- Max proficiencies between 0.5 and 1
- Max proficiencies between 0.75 and 1
- Max proficiency of 1

Convergence is immediate in the case where max proficiency for each agent is 1, but the more relaxed case sees a more varied strategy with a few agents not approaching giving full effort. For those agents in the last 30 iterations who gave less than .8 effort, the average maximum proficiency of the agent was .616.

### 1.3 Cost Reporting

Although not ideal, it may be possible to enforce some minimum proficiency level of an agent beyond .5 so as to ensure a faster convergence to full effort as an optimal strategy. However, even with a scaling factor of 10 the average reward for an agent in the full effort/truthful equilibrium was only around 12. This mechanism already isn’t entirely rational as it is possible for an agent to experience negative rewards, incurring any kind of cost would require a much larger scaling factor for the reward. The paper points out that all results from the paper hold when costs scale linearly to effort, but it would be more interesting to test under different, perhaps convex, cost curves.

However, computationally it becomes the case that either the costs outweigh the benefit of reporting truthfully and the optimal strategy is to give no effort or the reward scaling factor is so large that the costs are negligible resulting in the same results as the paper regardless of the cost curve. As seen from the results in Figure 1, the rewards have quite a lot of variance, so having a large scaling factor to account for costs could potentially lead to very high (and low) rewards. Altogether, it would be nicer to have a modified mechanism that accounted for costs irrespective of the paper’s main reward function.

#### 1.3.1 Take It Or Leave It Crowdsourced Elicitation

The Take It Or Leave It mechanism that was introduced in class involved asking for a cost from a user, sampling from a distribution, and conducting the survey from the user only if the sample was higher than the user’s cost. The mechanism was both truthful and individually rational. It is useful in our case in that we’d like to ask an agent for a cost to reimburse for reporting a task, but we would obviously want a truthful reporting of the cost. Using TIOLI, we introduce a modified crowdsourcing mechanism:

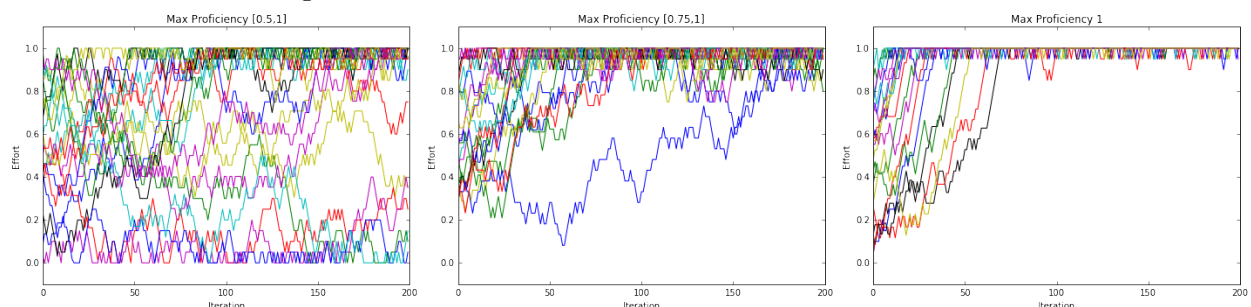
1. Distribute tasks to all agents in usual way
2. For each task, obtain a sample  $p$

3. For each agent, ask for a reported cost
4. Remove tasks for agents where reported cost is higher than corresponding  $p$
5. Remove tasks for agents who don't have a reference rater or sufficient non-overlapping tasks
6. Conduct crowdsourced mechanism and then add additional sample  $p$  as reward for each task

Finally, if we were to estimate the true signal for a task (e.g. true grade for a student from his peers' grades), we can first estimate the effort given by the agents via their reported cost. If there were some information about each agents' maximum proficiency, we could calculate the probabilities of their reports given the true signal of the task and see which signal is higher.

This mechanism would still incentivize telling the truth as reimbursing the cost gives no incentive to be untruthful and the reward function would still incentivize being truthful if there are other truthful agents. However, one's expected return for giving no effort is no longer 0 as an agent will receive a reimbursement for completing every task.

Figure 4: Take It Or Leave It Crowdsourced Mechanism



As Figure 4 shows, the agents are still generally progressing towards telling the truth when proficiencies are high, but in the case of only between 0.5 and 1 the strategies fluctuate heavily. Costs were set equal to effort level, and samples were drawn from a normal distribution with mean 2 and standard deviation 0.5. All other settings were the same as those from Figure 3.

If agent's utility functions aren't linear, they may be fine with gaining some utility for providing no effort. It is also possible that a task has no agents reporting it, though in very large settings this is unlikely. The choice of the distribution becomes very sensitive, but in general the results looked similar with various price distributions (even fixed price) above the maximum cost. We think this is preferable to simply scaling the reward factor of the original mechanism as there are more reasonable bounds on the possible reward given out to reimburse costs whereas as seen in the original computational results, rewards can be quite varied.

### 1.3.2 Differing Cost Curves

Whereas previously the choice of the scaling factor for rewards resulted in either dominating the costs or being dominated and subsequently the optimal strategy went to full or no effort, now with the take it or leave it mechanism varying proficiency curves can be more closely examined.

Figure 5: Cost Curve Effects

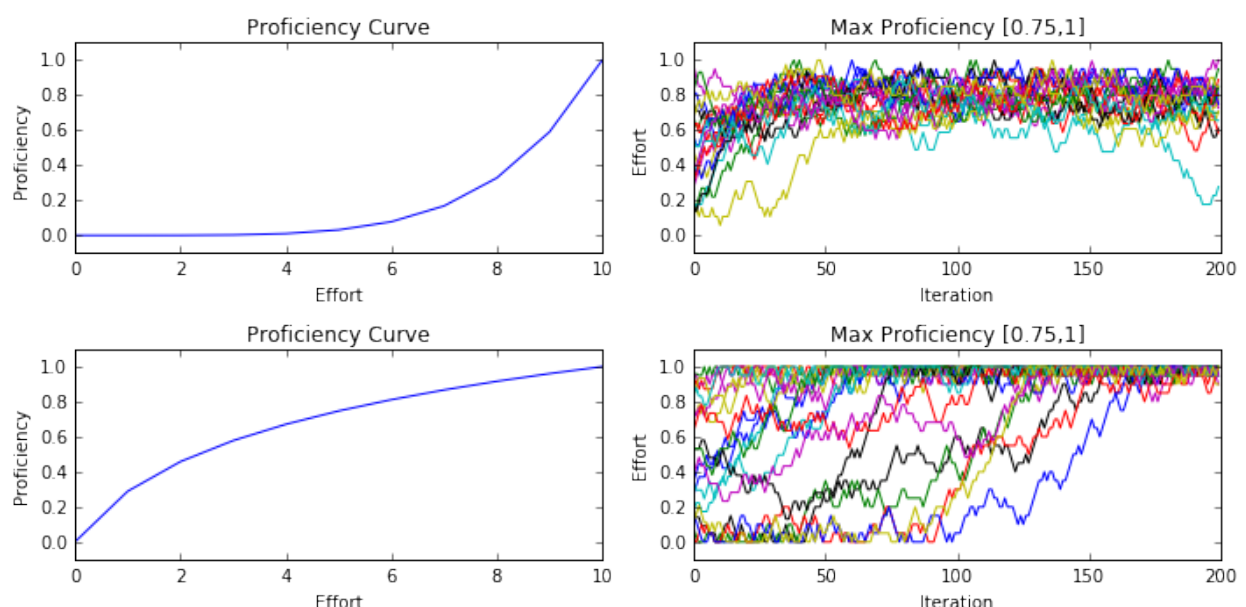


Figure 5 shows a new proficiency curve corresponding to the effort given (and consequently the cost which are linear to effort). We see that there is no longer a convergence to fully telling the truth in the convex case, whereas the convergence seems to happen in the concave case albeit slowly.

## 2 Peer Prediction

We recall the main assumptions made in the paper:

1. Conditional on the products type, raters signal sare independent and identically distributed, i.e. raters have common prior
2. Raters have common knowledge of the probability of reporting a signal given a type of a product.

The objective of this project being to relax some of these assumptions, we implemented an agent class in python that would allow for different prior distribibutions as well as personal signal probability given the particular type of a product. We also implemented the various score functions mentioned in the paper, i.e. Quadratic, Spherical and Log scoring rules.

We then wanted to verify the main proposition of the paper:

For any mapping  $r$  that assigns to each rater  $i$  a reference rater  $r_i = i$ , and for any proper scoring rule  $R$ , truthful reporting is a strict Nash equilibrium of the simultaneous reporting game with transfers  $i$ .

To do so, we simulated 1000 experiments where 2 agents who satisfy the assumptions presented above, reports 1000 signals over two types. We fix agent 2 to always be truthful and let agent 1 be truthful, deceitful or even adopt a randomised strategy (0.5, 0.5). As we can see on figure 4, it seems to always be better to report truthfully, even if it is only slightly more advantageous than to report using a randomised strategy. This last statement could also be invalidated with further testing and different randomised strategies.

We also noted an interesting results when trying to relax the second assumption stated above. For Proposition 1 to hold, we in fact need not only that raters have common knowledge of the probability of reporting a signal given a type of a product, but also that these distributions are to be the same. To validate this new assumption, we simulated the same experiments as before, but with agents having different distributions of perception, which is in fact close to the proficiency consideration of the crowd-sourcing paper. In one case, we used an agent that was rather undecided when receiving a signal, having a 50% chance of correctly reading the signal. In the other, we designed an agent that was very bad at identifying the type for which the second agent is actually quite good at, and keeping the second type distribution the same. Figure 6 shows that to always lie is a better strategy for agent 1, if he knows that he tends to read signals worse than his refering agent !

Next steps of this part of the project woudl be to focus on this issue as well as relaxing the first assumption of common priors over types.

## 3 Results

### 3.1 Peer Prediction

Figure 6: Computational verification of Property 1

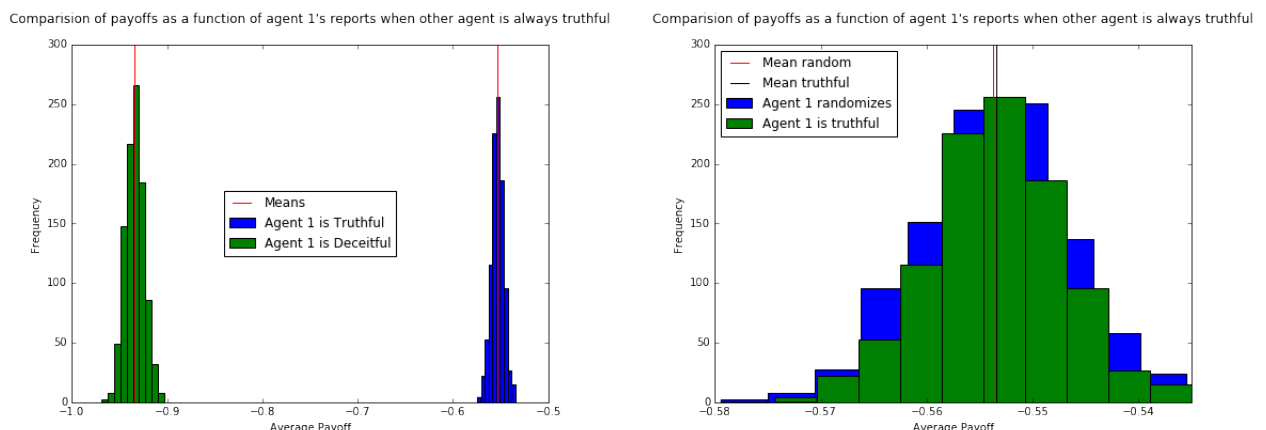


Figure 8: Relaxing Assumption 2

