**Milestone Report**

Subcontract Number: 16102-172910-QS

Subcontractor’s Name: \_Harvard University\_\_\_\_\_\_\_\_\_

Address: \_\_\_33 Oxford St., Cambridge MA 02138\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of PI: \_Yiling Chen and David C. Parkes\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Reporting Period: \_\_August 7, 2017 – December 31, 2017\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date Report Prepared: \_January 5, 2018\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Machine-Aided Human Forecast Module – Crowdsourcing Compilation

1. A description of a formalism for Human-Machine Collective Intelligence

We focus here on the human-facing aspects of the machine-aided human forecast module. We propose that the forecast module should:

1. use a simple wagering mechanism to elicit predictions and signals of confidence from individuals,
2. place participants into teams, to facilitate cohesive discussions and information exchange and to motivate,
3. facilitate structure around the reasoning provided for a forecast,
4. match IFPs to teams,
5. provide an open-text discussion forum for members of a team,
6. use peer prediction to score contributions and motivate participants, and
7. provide a team leaderboard (based on wagering), an individual leaderboard (based on wagering), and badges (based on peer prediction scores).

For the *wagering mechanism*: we’ll propose a *No Arbitrage Wagering Mechanism* (NAWM) to elicit probabilistic forecasts and allow participants to signal their confidence about their predictions through their wagers.[[1]](#footnote-1) The prototype of the wagering mechanism is provided in point 2, and the detailed description is provided in point 3.

For *teaming*: we propose two treatments. In treatment one, we have *specialized teams,* and evaluate the goodness of a team based on how the maximum sum ability of team members across all skills. In treatment two, we have *diverse teams*, and evaluate the goodness of a team based on how many skills can be covered, when each individual can cover at most one skill. Some more detail is outlined in point 3 below.

For allowing for *reasoning patterns* to be described around a forecast: we propose to allow forecasters to indicate the kinds of reasoning provided, and optionally to structure their response. Some more detail is outlined in point 3 below.

For *matching IFPs to teams*: we propose to make ~20% of the assignments at random and ~80% according to match with the skills of a team, while also respecting balance across teams and seeking to assign each IFP to ~50% of the teams. Some more detail is outlined in point 3 below.

The *open-text discussion forum* can be of a standard form, for example a fully-featured version would be similar to Slack. It should be restricted to team members, and allow team members to “poke” people on the team and follow discussions, as a way to pull them back into the platform.

For the use of *peer prediction,* we propose to use the *Correlated Agreement mechanism[[2]](#footnote-2)* as a way to activate contributions to the site before IFPs are resolved-- both IFP predictions themselves, as well as feedback that could be elicited on contributions of other team members (e.g., credibility of a source, agreement on a fact, etc.). Some more detail is outlined in point 3 below.

For *leaderboards*: these should be (i) team-based, with a team’s score coming from Brier scoring applied to a team’s aggregate team prediction made each day. This team prediction is either the median of all team members’ standing predictions or a weighted combination of the standing predictions from each team member, with the weights proportional to their wagers; (ii) individual-based, with an individual’s score coming from the NAWM. For *badging*, we propose that badges be provided for different levels of contributions (measured according to blend of quantity of contribution and peer-prediction score on contribution), with badges tiered according to absolute scores (and new badges introduced as necessary to allow for continued motivations).

1. An Algorithm and Prototypes for Machine-aided Human Forecast

The prototype for NAWM is produced as a result of the human-subject experiments that we have conducted on wagering mechanisms. The code of the prototype is available at the Github repository:

<https://github.com/yangl0320/wager_exp.git>

1. A Description of the Elicitation Environment and Mechanism

We provide short memos to describe each of the following design components:

1. The design of the no-arbitrage wagering mechanism (NAWM)
2. The design of an algorithm for team assignment
3. A method to promote structured reasoning
4. The design of an algorithm for matching new IFPs to teams
5. The design of the peer-prediction mechanism

Memo for Implementing the Wagering Mechanism

**Timeline**

***Start of the day:***

1. Some IFPs close, some new IFPs open.
2. Each user gets fresh points as their daily budget.
   1. The number of points depends on the number of open IFPs. For example, we give 5 points for each open IFP - suppose one day there are 20 open IFPs, we will allocate each forecaster/user 100 points that day.
3. Users can see their rewards or scores they can get on different outcomes of a particular IFP they participated yesterday or earlier, based on existing forecasts made by other users.

***During the day:***

1. Users login in and select IFPs to predict, they can select any open IFP to make a new forecast or update their old forecast. They have to allocate their daily budget to IFPs as their wagers on these IFPs. They are allowed to allocate 0 point, but neither negative points, nor exceeding their daily budget.
2. If some users don’t login or don’t update their forecasts, their forecasts and point allocations made by yesterday will be carried over. If they don’t use up all their daily budget, the rest of the budget should NOT be carried over the next day.
3. Similar to the GJOpen project, the above procedure for a particular IFP only starts after the user made her/his first forecast.

***End of the day:***

1. Our system collects the forecasts and point allocations that users made during the day and the forecasts and point allocations of yesterday if some users didn’t update their decisions during the day.
2. For each IFP, our system runs a no-arbitrage wagering mechanism (NAWM) for all possible outcomes of the IFP. The inputs are the **forecasts** and **points** made by users on this IFP and the outputs are the rewards (scores) of all users under different possible outcomes of the IFP. Users are allowed to check their own results the next day, corresponding to the 3rd item in “Start of the day”.
3. When the outcome of an IFP is realized someday, users’ scores are updated according to the result of the NAWM running under this outcome. This is a one time computation.
4. **Forecast aggregation:** our system will aggregate the forecasts at the end of each day. So far in our experiment, we tested simple points-weighted averaging method. This works as follows: suppose for a particular IFP, we have received a set of (prediction, points): (p\_1, points\_1), (p\_2, points\_2), …, (p\_N, points\_N). Then the aggregation works as follows:

p\_aggregate = p\_1 \* (points\_1/sum\_i points\_i) + ....+p\_N \*(points\_N/sum\_i points\_i)

**Details of running NAWM**

***Partition of forecasters for participating the wager games:***

We are going to partition the forecasters into multiple groups, and run the wagering mechanism over each group separately. We propose to use the following method[[3]](#footnote-3):

1. We rank forecasters w.r.t. their forecasts either in an ascent order or a descent order.
2. Then we partition these ranked users into 10 tiers (the number of tiers equals the size of a group) based on the above ranking from the top to the bottom, e.g., first 100 users into tier 1, second 100 users into tier 2, …, last 100 users into tier 10 when there are 1000 users in total.
3. From each of the 10 tiers, we randomly pick up a user and group these 10 users into one group. These users are never put back. We repeat this step until all users are group into exactly one of the small groups.

The above partition ensures that the population for each wager game is diverse.

***Calculating the wager scores for each forecaster:***

1. At the end of the day, for a particular IFP, our system collects the forecasts and points forecasters made or carried over on this IFP. For example, we have 1000 probability forecasts and points allocated to the IFP from 1000 different forecasters.
2. We partition the users into groups of a small size, e.g., 10 forecasters a group. For each group, we run an individual NAWM and reward (score) forecasters correspondingly.[[4]](#footnote-4)

***Feedback for users:***

1. Forecasters should be familiar with the idea of the NAWM before they make forecasts and point allocations.
2. Forecasters should know they are grouped into small ]groups when we run NAWM but not how we group them nor the result of grouping.
3. Forecasters should have a NAWM calculator for each IFP forecasting page, where they can specify the forecasts and points allocations for at least three players and observe the outcome of the mechanism under all possible outcomes of an event, e.g. for a binary event, users could observe the rewards (scores) of each player if the outcome 1 happened and the rewards (scores) if outcome 2 happened. We suggest either add such a calculator to the right hand side blank space under the prediction tab or have a ``help’’ tab for the calculator.
4. Forecasters should be able to check their rewards or scores they could get on different outcomes of a particular IFP they participated yesterday or earlier.
5. Forecasters should be able to check their actual rewards or scores when the outcome of a IFP they participated was realized.

Memo for the Assignment of Forecasters to Teams

We propose to assign 50% of participants to a “specialized teams” and 50% to “diverse teams”. Both normal participants and Turkers will be placed onto teams.

Let **v\_ij** denote the expertise that individual i contributes to skill j (skill could be “middle east” or “statistical analysis”). We suggest these values be inferred from the historical performance of participants on IFPs, together with an analysis of questionnaire responses (e.g., with PCA used to pull out additional dimensions). We do not consider in this memo how these values are assigned. The v\_ij values should be normalized so that they have the same mean/variance across skills.

Let **Q** = desired team size (e.g., 8).

Let **Q\_t** = desired number of turkers per team (e.g., 3).

**Step 1:**

Randomly rank order the normal participants, and assign round robin to D, S, D, S, D, … (Diverse, Specialized, ...) Randomly rank order the Turkers, and assign round robin to D, S, D, S, D, …

**Step 2:**

2a. For the participants in the diverse split, let **N\_d** = min ( floor( W\_t / Q\_t), floor(W\_n / (Q-Q\_t)) , where **W\_t** is the number of turkers in this split, and **W\_n** is the number of “normals” in this split. **N\_d** is the number of diverse teams we will form. It is defined so that there will be at least Q\_t and at least Q total participants per team.

We think of a good, diverse team as one where each person brings a skill to the table. For example, a team where person 1 has skill A and person 2 skill B is better than a team where person 1 has skills A and B, person 2 nothing. We evaluate the degree of diversity of a team T as

max\_x \sum\_{i\in T} \sum\_{j\in M} x\_{ij} v\_{ij}

s.t. \sum\_i x\_{ij} <=1, forall j

\sum\_j x\_{ij} <=1, forall i

x\_{ij} \in {0,1}

where **M** is the set of skills and **T** is the set of individuals on a team.

Given this, we solve the following integer program to define diverse teams:

x\_{ijk} do we use skill j of person i on team k

\sum\_j x\_{ijk} ... is person i assigned to team k?

Introduce a dummy skill, j0. Everyone has zero value for this.

For the people in the diverse split:

max \sum\_{i \in people} \sum\_{j\in skills} \sum\_{k \in teams} x\_{ijk} v\_{ij}

s.t. \sum\_{jk} x\_{ijk} = 1 , for all i {assign each person to a team}

\sum\_i x\_{ijk} <= 1, for all j \in real skills, for all k {credit at most 1 skill per team}

\sum\_{i\in Norms} \sum\_j x\_{ijk} >=Q-Q\_t, for all k {enough “normals” on a team}

\sum\_{i\in Turkers} \sum\_j x\_{ijk} >= Q\_t , for all k {enough Turkers}

x\_{ijk} \in {0,1}

2b. For the participants in the specialized split, let **N\_s** = min ( floor( W\_t / Q\_t), floor(W\_n / (Q-Q\_t)) , where **W\_t** is the number of turkers in this split, and **W\_n** is the number of “normals” in this split. **N\_s** is the number of specialized teams we will form. It is defined so that there will be at least Q\_t and at least Q total participants per team.

We think of a good, specialized team as one where the team is collectively good at one thing. For example, a team where person 1 has skill A and person 2 skill B is worse than a team where both 1 and 2 have skill A. We evaluate the degree of specialism of a team T as

max\_{j\in M} \sum\_{i\in T} v\_{ij}

where **M** is the set of skills and **T** is the set of individuals on a team.

Given this, we solve the following integer program to define teams amongt the specialized split:

W\_kj : the value of team k on skill j

W\_k: the value of team k

Q\_{jk} : idicator variable , is skill j the best for team k

X\_{ik}: is i assigned to team k?

max \sum\_k W\_k

s.t. W\_{kj} <= Q\_{jk} M {only allow W\_{kj} to be non-zero if Q\_{jk} is 1}

W\_{kj} <= \sum\_i x\_{ik} v\_{ij}

W\_k = \sum\_j W\_{kj}

\sum\_j Q\_{jk} =1 ,for all teams k {pick the best skill}

\sum\_k x\_{ik} = 1 , for all people i {assign each person to a team}

\sum\_{i\in Normals} x\_{ik} >= Q-Q\_t , for all k {enough “normals” on each team}

\sum\_{i\in Turkers} x\_{ik} >= Q\_t , for all k {enough Turkers}

x\_{ik}, Q\_{jk} \in {0,1}

W\_k, W\_{kj} >= 0

Memo for UX design to Promote Structured Reasoning

This has two elements:

1. a plain text box to enter a reason for an IFP prediction, with a “+” to click and bring up an additional box (to optionally allow a reason to be broken into pieces)
2. a set of toggle buttons next to the plain text box, allowing for a user to select 0 through k (where there are k buttons in total) of the following: **fact, source, analysis, related event.**

Memo for the Matching of IFPs to Teams

We expect IFPs to arrive in batch, so that an optimized decision can be made about how to match them to teams.

We propose to assign each IFP to some fraction of teams, say **f = 0.5.** This is set to prevent the teams becoming overloaded. Suppose there are 100 teams, and thus each IFP would be assigned to 50 teams (with f= 0.5).

We propose that some fraction, say **r = 0.2**, of the 50 assignments for an IFP be made at random. This will serve the purpose of providing an experimental control. The rest of the 50 assignments, say the other 80%, will be provided according to team expertise but while recognizing the need to keep the amount of assigned work relatively balanced across teams. We think about a “good” assignment of an IFP to a team as one for which the total “compatibile” expertise provided by team members is maximized.

Let **N** denote the set of teams (considering both specialized and diverse). Let **L** denote the number of IFPs currently assigned that have not closed. Let **L\_i** denote the number of IFPs that are currently assigned to team i \in N.

Let **M** denote the set of new IFPs to assign.

Let **w\_{ij}** denote the “value” (or qualification) of team i for IFP j. We define this as

w\_{ij} = sum\_{k\in S\_j} \sum\_{u \in T\_i} v\_{uk}

where **S\_j** is the set of skills that are defined to be relevant for IFP j (this needs to be defined in some way, e.g. via simple NLP methods), and **T\_i** is the set of individuals on team i. v\_{uk} is the value of individual u for skill k, and could be as defined above when discussing the assignment of individuals to teams.[[5]](#footnote-5) That is, we think about the value of assigning an IFP to a team as being the sum expertise that is relevant to the IFP provided by people on the team. Note: this expression may need to be tweaked, because it might overly favor diverse over specialized teams in the case that an IFP has multiple relevant skills.

**Step 1**. For each of the M new IFPs, make a random assignment of fraction r of its total assignments. In particular, assign **each** IFP to **k1 = ceil( r \* f \* N)** teams.

For this, place teams into random priority order, **o**. Assign the new IFPs round robin according to this order, i.e. if k1= 2 and M = 4, then form sequence IFP1 IFP1 IFP2 IFP2 IFP3 IFP3 IFP4 IFP4, and assign these round robin to teams according to team priority order **o**.

**Step 2.** Expertise -based assignment

Now assign each of the M new IFPs to **k2 = ceiling( (1-r) \* f \* N)** teams, according to the expertise of the teams but also considering the need for balance.

Let **L\_i denote the updated number of IFPs** assigned to team i, including new random assignments.

To reason about the ideal *balanced* solution (that would ignore expertise), first solve:  
 the following IP:

x\_{ij} : is task j assigned to team i?

y : max assignment over all teams

z: min assignment over all teams

min y-z

s.t. y >= L\_i + \sum\_j x\_{ij} , for all i {y is the max assignment over teams}

z <= L\_i + \sum\_j x\_{ij}, for all i {z is the min assignment over teams}

\sum\_i x\_{ij} = k2 , for all j {each IFP is assigned k2 times}

x\_{ij} = 0 , if IFP j has already been randomly assigned to team i

x\_{ij} \in {0,1}

z, y >=0

This gives the optimal balanced assignment, in the sense that it would best minimize the difference between max and min assignment. z is the resulting minimum number of IFPs assigned to teams.

We use this solution to guide the expertise-based IFP assignment. For this, define **z\_floor** = z – 1 and **z\_ceil** = z + 1. We will insist that the total assignment to each team is between z\_floor and **max(L\_i, z\_ceil)**. This allows for some flexibility in considering expertise, while keeping reasonably well to balance requirements.

For this, we solve the following integer program.

Let **x\_ij** \in {0,1} denote whether task j is assigned to team i.

max\_x \sum\_{1=1}^N \sum\_{j=1}^M w\_ij x\_ij

s.t. z\_floor <= \sum\_{j=1}^M x\_{ij} + L\_i <= max(L\_i, z\_ceil), for all i {balance}

\sum\_{i=1}^N x\_{ij} = k2, for all IFPs j {demand}

x\_{ij} = 0 , if IFP j has already been assigned to team i

x\_ij \in {0,1} , for all teams i, for all IFPs j

By construction, this IP is always feasible.

Memo for the Design of a Peer Prediction Mechanism

We propose to use the Correlated Agreement (CA) mechanism to score contributions by forecasters of the following kind:

* IFP predictions before the IFP is resolved
* Feebdack on the contributions of others; e.g., up/down votes on IFP predictions, up/down votes on reasons provided

The CA mechanism does not need to be used just to score binary feedback. For example, if we wanted a multi-part feedback on a reason (e.g., “helpful” “correct” “detailed” etc then this could also be handled).

A peer prediction (PP) method is a method to motivate participants to (i) invest effort in the context of information elicitation, and (ii) provide their true belief , in a setting where there is no verification of the information. That is, the role of PP is analogous to that of wagering but for a setting where there is no later “reveal” of the true outcome.

For the context of HFC, PP can be used both to score IFP predictions before the event is realized and to score, for example, up/down votes on the reasons of others or IFP predictions of others.

The chief advantage of the CA mechanism relative to many other PP mechanisms is that it provides the following two properties

1. expected score from truthful participation given that others are truthful is at least as much as from any coordinationed play
2. expected score from truthful participation given that others are truthful is strictly better than from any unininformed play (e.g., constant inputs, or random inputs)

The CA mechanism requires as input the *correlation structure* between reports of two participants. That is, does “up vote” tend to be positively or negatively correlated with “up vote”, and similarly for other signals. This can either be defined when constructing the mechanism or learned from reports (without compromising incentives, just deferring scoring until the correlation structure is well enough understood).

For IFPs, we could first of all discretize the prediction space, e.g. into 10 buckets, and then provide the correlation structure in that 10x10 space (for example, we have fitted a model to do this based on GJopen data).

For each report provided by a forecaster, the CA mechanism works by “scoring” this report against that of another forecaster for the same task but then subtracts the score for this forecasters report on some second task and the report of another on a third task. That is, the score is X(a1,a2) – X(b1,c2) where X is the score function and a, b, .. indicates the task and 1, 2, .. the forecaster. In the binary case, such as up-vote and down-vote then we can generally think about X(.,.) as 1 when the inputs are the same and 0 otherwise.

The CA mechanism defines X to be 1 when the two signals are positive correlated and 0 when they are negatively correlated.[[6]](#footnote-6)

1. Yiling Chen, Nikhil R. Devanur, David Pennock, and Jennifer Wortman Vaughan. [Removing Arbitrage from Wagering Mechanisms](http://dl.acm.org/authorize?6925642). In Proc. of the 15th ACM Conference on Economics and Computation (EC), Palo Alto, CA, June 2014.  [↑](#footnote-ref-1)
2. Victor Shnayder, Arpit Agarwal, Rafael M. Frongillo, and David C. Parkes. 2016. “Informed Truthfulness in Multi-Task Peer Predictions.” In Proceedings of the 17th ACM Conf. on Economics and Computation (EC16), Pp. 179-196. [↑](#footnote-ref-2)
3. We haven’t implemented this, but could help code it up. [↑](#footnote-ref-3)
4. Here we need to call the no\_arbitrage\_wagering function coded in the python script. [↑](#footnote-ref-4)
5. Note that this could be updated over time, to allow for an adaptive approach to assignment of IFPs to teams [↑](#footnote-ref-5)
6. For more information: Victor Shnayder, Arpit Agarwal, Rafael M. Frongillo, and David C. Parkes. 2016. “Informed Truthfulness in Multi-Task Peer Predictions.” In Proceedings of the 17th ACM Conf. on Economics and Computation (EC16), Pp. 179-196. [↑](#footnote-ref-6)