The Effectiveness of Peer Prediction in Long-Term Forecasting

Debmalya Mandal
Columbia University

Goran Radanovic
Max Planck Insitute of Software Systems

David C. Parkes
Harvard University

Abstract

In human forecasting, proper scoring rules are used to elicit effort in providing accurate probability forecasts of future events. A challenge, though, is that users do not receive feedback about their forecasts until the outcomes are realized. Nor is it clear whether these schemes are effective in motivating continual attention, and updating forecasts on difficult or dynamically changing problems, for which there is a continuous inflow of new information over time.

Through a large-scale experiment on Amazon Mechanical Turk (MTurk), we investigate whether peer prediction methods can be used to complement methods of proper scoring, and improve engagement of users and ultimately the quality of forecasts. Peer prediction provides immediate feedback, by comparing one forecaster's prediction with that of another, this feedback provided as rank placement or through incentive payments. One of a very small number of experimental studies into peer prediction, ours is the first to test peer prediction in this hybrid role.

We show that providing daily feedback through peer prediction has a significant effect in increasing engagement with the forecasting platform. Moreover, a hybrid scheme that combines scoring rules with peer prediction feedback (via rank feedback) is, together with the basic scoring rule method, generally the best for accuracy. Since the hybrid scheme also improves user engagement, this suggests that the hybrid scheme would provide the best accuracy for longer term forecasting events.

1 Introduction

In contrast to fully automated systems, many application domains are expected to rely on combined human-AI intelligence, a good example of which would be forecasting systems. Recently, the Intelligence Advanced Research Projects Activity (IARPA)[1] organized the Hybrid Forecasting Competition[2](HFC) program to promote the next generation of geopolitical forecasting systems based on human-AI collaboration: human experts provide information that is supported by AI tools, such as aggregation methods.

One of the key challenges in forecasting uncertain events is the time changing nature of their underlying best estimates, requiring frequent updating of existing information content upon which forecasts were made. The new information content is often obtained by querying multiple human experts, and to ensure its quality, these experts can be paid in proportion to the marginal information gain they bring to the forecasting system.

One possible information that experts can provide to the system are their own forecasts, which the system can aggregate to produce a final estimate. A standard way to measure the quality of experts' forecasts, and thus assign scores, is through *strictly proper scoring rules* - a class of scoring techniques that are maximized in expectation for the best prediction. A well known example of a strictly scoring rule is Brier score (Brier, 1950).

While such scoring techniques are strictly proper (accuracy rewarding), they do not provide any immediate signal to experts regarding their performance, lacking a form of implicit incentives in interim periods, before the event is realized. This kind of interim incentive might effectively encourage an expert to update their (low quality) forecasts more frequently since they indicate the quality of the expert's current estimates, which is in turn correlated to the expert's score. As empirically shown by Ungar et al. (2012), the number of forecasts that an expert makes tends to correlate with an expert's performance, and this is in part explained by up-to-date forecasts being more accurate.

To provide interim incentives, the system can score experts based on the consistency of experts answers—such an approach leads to *peer-prediction mechanisms*. Peer-prediction mechanisms can provide more frequent rewards than scoring rules, but the properness of peer-prediction scores depends not only on an individual participant's subjective beliefs, but on the strategy that all experts adopt. Fortunately, a wide range of manipulation strategies can be avoided with careful design.

The third possibility is a hybrid incentive approach that seeks to combine the best of both worlds, that is, in

[1] https://www.iarpa.gov/

[2] https://www.hybridforecasting.com/

which a scoring rule is supported by a peer-prediction mechanism. By carefully tuning the weights put on scores derived from the scoring rule and scores derived from the peer-prediction, the hybrid approach can control the theoretical non-manipulability of the overall incentive with the desired strength of interim incentives.

In this paper we aim to answer two main questions:

1. *Which incentive approach leads to the highest updates in forecasts?*

2. *Which incentive approach leads to the highest prediction accuracy*?

Additionally, we are interested to see whether a particular kind of question responds better to a particular kind of incentive approach. In this regard, we consider two different partitions of the set of questions. First, we consider partitioning the questions based on information inflow — *static* (information inflow from experts and other relevant factors are expected to be low), and *dynamic* (information inflow is expected to be high). Second, we consider a partition based on the hardness of the question— *hard* (questions that are harder to predict and the average accuracy under the baseline scoring rule treatment is below a given threshold), and *easy* (questions that are easier to predict and the average accuracy under the scoring rule treatment is above a given threshold).

To answer these questions, we conducted a large scale experiment on Amazon Mechanical Turk (MTurk) over the course of several days. Our main result is that peer-prediction incentives, whether in monetary or in non-monetary form such as rank, signficantly improves engagement with the forecasting platform compared to basic, scoring rule based incentives. Furthermore, we use various aggregators to evaluate the accuracy of aggregated forecasts in different treatments, and find that two treatments, the basic *scoring rule* and a *hybrid scheme* which, in addition to the scoring rule based payments, sends daily non-monetary feedback to the users, perform best for both dynamic and easy questions, and signifcantly outperform the other treatments for some aggregators. Moreover, because the new hybrid scheme improves user engagement compared to the scoring rule treatment, we believe that such hybrid scheme will outperform basic scoring rule based incentive schemes for longer term forecasting events.

To the best of our knowledge, this is the first long-term experiment to emprically test the performance of hybrid incentive schemes. Here we focus on forecasting problems, but we believe that the results are likely to hold for other elicitation settings in which elicited information varies over time. Of independent interest, we show how to elicit forecasts using a detail-free peer-prediction, which is, to our knowledge, the first results of such type reported in the literature.

1.1 Related Work

Our work is most closely related to the literature on information elicitation, out of which we emphasize two incentive approaches that are most relevant for this work.

Scoring rules. When the mechanism designer has access to the ground truth, she can use the ground truth to evaluate the users' responses. To incentivize truthful reporting of probability forecasts, one can use *strictly proper scoring rule* (Brier, 1950; Gneiting and Raftery, 2007)— any rational agent who faces a strictly proper scoring rule will always report her probability assessment of an event truthfully to the mechanism designer to maximize her expected payment. Examples of strictly proper scoring rules include *quadratic scoring rule* and *logarithmic scoring rule*. We use a variant of the quadratic scoring rule, which we define in later sections.

Peer prediction. In contrast to incentives based on direct verification, peer prediction mechanisms construct incentives by comparing a user's response with those of their peers (Miller, Resnick, and Zeckhauser, 2005; Prelec, 2004; Jurca and Faltings, 2009; Witkowski and Parkes, 2012; Dasgupta and Ghosh, 2013). These techniques assume that users have correlated information. However, unlike scoring rules, they are applicable even when the the ground truth is not known to the mechanism, e.g., because the elicited information contains probability estimates about an event that realizes in a distant future. In recent years, such methods have been studied in several domains, including massively open online courses (MOOCs) (Shnayder et al., 2016), for eliciting feedback on local places in a city (Mandal et al., 2016) and also in the context of collaborative sensing platforms (Radanovic and Faltings, 2015). We use *Correlated Agreement* (CA) of Shnayder et al. (2016), which has provable guarantees on collusion resistance for a wide variety of reporting strategies, and we adopt it to our forecasting setting.

Experimental work. While different incentive mechanism designs have been experimentally tested, especially for the purposes of crowdsourcing (e.g., see (Shaw, Horton, and Chen, 2011)), much of the work on hybrid designs that combine peer prediction with gold standard incentives is grounded in theory and simulations (Gao, Wright, and Leyton-Brown, 2016; Goel and Faltings, 2019). This is not surprising given that experimental work on peer prediction has only focused on a few particular cases. These include: (a) Garcin and Faltings (2014) showing that peer prediction can elicit forecasts which accuracies are comparable to those elicited via *prediction markets*[3]; (b) Gao et al. (2014) showing that collusion can occur if a small group of users is repeatedly asked to report their private information; (c) (Radanovic, Faltings, and Jurca, 2016) showing that robust peer prediction designs perform well in peer grading. More general overview of the literature on incentive mechanism design can be found in Faltings and

[3]Prediction markets require the ground truth for scoring, and some designs are closely related to strictly proper scoring rules Hanson (2012).

Radanovic (2017).

2 Experiment

We ran our experiment on Amazon Mechanical Turk (MTurk) and it consisted of two HITs: a *recruitment HIT* (restricted to US only) and a *forecasting HIT*, where the recruited workers were participating in the actual study. We use two separate HITs because we need to send daily feedback to the workers (monetary and / or non-monetary) and the MTurk platform does not allow to send bonus payments on an ongoing HIT. For the forecasting HIT, we use the HIT ID and the assignment id of the recruitment HIT to provide bonus payments to the workers. Our experiment consists of four treatments:

1. **Scoring Rule** (SR): The workers are paid according to the Brier Scoring rule once the outcomes of the events are realized at the end of the study. We also provide a daily reminder, suggesting workers should come back to the platform and update their predictions if they have any new information.
2. **Peer Prediction** (PP): The workers are paid daily according to a peer prediction method. We use the correlated agreement mechanism (Shnayder et al., 2016) to compute daily bonus payments. The details of the mechanism is discussed later.
3. **Scoring Rule + Peer Prediction (Rank)** (SR+PPRank): The workers are paid according to the Brier Scoring Rule once the outcomes of the events are realized at the end of the study. Each day we compute the peer prediction score based on the latest predictions of the users. Instead of providing users with actual bonus payments we provide them ordinal feedback, i.e. mentions which quartile the user belongs to in terms of her daily peer prediction score (top 25%, 25%-50%, 50%-75%, or bottom 25%). [4]
4. **Scoring Rule + Peer Prediction** (SR+PP): The workers are provided with two types of bonus payments: (a) they are paid according to the Brier Scoring Rule once the outcomes of the events are realized at the end of the study, and (b) they are provided daily bonuses based on the peer prediction score computed using their latest predictions. We normalize the payments so that the expected payments from (a) and (b) are the same.

2.1 Experimental Workflow

The recruitment HIT was posted on October 6, 2018, while the forecasting HIT was posted on October 7 and was online till October 13. Overall, 945 workers out of the 1400 workers who participated in the recruitment HIT signed up for the forecasting HIT, but we considered only 891 of the 945 workers— those who joined the forecasting HIT on October 7th and were present for seven days. The following table shows the breakdown of the 891 workers across the four treatments.

Treatment	# Workers
Scoring Rule	207
Peer Prediction	220
Scoring Rule + Peer Prediction (Rank)	238
Scoring Rule + Peer Prediction	226

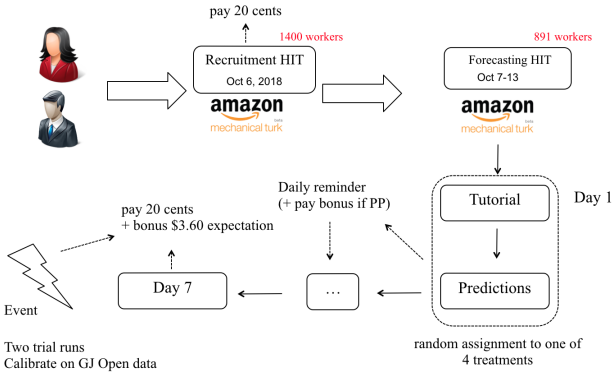


Figure 1: Experimental Workflow

Figure 1 shows the details of the experiment including the payments. We asked the workers to provide forecasts on 18 different questions. There were three different categories of questions: (a) Sports, (b) Politics and Economy and (c) Entertainment, with six questions from each category. The supplementary material includes the questions along with their outcomes.

For our analysis, we consider two types of partitions of the 18 questions. First, we had nine *dynamic* questions and nine *static* questions. A question is dynamic if there is a continuous inflow of information for that question over the span of seven days. An example of a dynamic question is:

- "Will the price of Bitcoin in USD on Sat Oct 13th (EST time zone) be, at any point of the day, above 6500? "

On the other hand, the workers do not receive new information for static questions, e.g.:

- "Will D.C. United win the D.C. United vs. FC Dallas soccer game (Major League Soccer) on Sat Oct. 13th?"

In addition, we also partition the questions based on how hard they are to predict. An example of a *hard* question is

- "Will the price of Bitcoin in USD on Sat Oct 13th (EST time zone) be, at any point of the day, above 6500?"

This is a hard question because leading to the week of October 7th, the price of Bitcoin in USD has been

[4] We decided to provide ordinal feedback instead of the actual peer prediction score because the scores are not converted to payment for this treatment and the workers can assess how well she is doing relative to the whole population from the rank / quartile.

around 6500. In general, different forecasters might have different opinions about what makes a question easy. So we computed the average accuracy of all the questions after they were realized and decided that questions with average accuracy above 0.71 will be labelled as easy. Section 7.2 in the supplementary material justifies the choice of the threshold. We had seven *hard* questions and eleven *easy* questions.

2.2 Payment Scheme

For each question, we provide a base payment of \$0.20 for the recruitment HIT and a base payment of \$0.20 for the forecasting HIT. The payments of the four treatments are based on the Brier scoring rule Brier (1950) and the correlated agreement (CA) mechanism Shnayder et al. (2016). In particular, the payment for SR is completely determined by the Brier scoring rule. The payment for PP is determined by the CA mechanism. The treatment SR+PPRank uses the Brier scoring rule for payment, but uses CA to provide non-monetary feedback. The treatment SR+PP uses both the Brier scoring rule and the CA mechanism. Next, we explain the Brier Scoring Rule and the CA mechanism.

Brier Scoring Rule To explain the payments obtained via the Brier scoring rule, let us fix an event and suppose that on day j a worker has a forecast of f_j for the outcome of that event. In case the worker does not enter a forecast on a particular day, we carry forward her forecast from the previous day. Once the outcome of the event is realized, the payment to the worker is given as $\sum_{j=1}^{d} s_j$. Here s_j is computed using the Brier scoring rule as :

$$s_j = \begin{cases} 1-(1-f_j)^2 & \text{if the outcome is one} \\ 1-f_j^2 & \text{if the outcome is zero} \end{cases}$$

To summarize, once the outocme of the event has been realized, the worker is rewarded for each of her daily forecast. In case the worker comes back to the platform several times on a day we pick the latest forecast to compute the reward.

CA Mechanism Unlike the Brier scoring rule which determines the score (payment) for each question separately, CA uses the fact that each user is assigned multiple questions. In the absence of the actual outcome, CA computes the payment (score) depending on its reports and the reports of its peers. The main idea behind CA is that it rewards agreement on the same question and punishes agreement on two separate questions. For an agent p, the reward for her response on question j is computed in the following way.

1. Pick two questions t' and t'' different from j (penalty questions) such that p has completed question t'.
2. Randomly select an agent $q \neq p$ such that q has completed question j and t''.
3. Let the reports of agent p on questions j and t'' be r_p^j and $r_p^{t'}$ respectively. Let the reports of agent q on questions j and t'' be r_q^j and $r_q^{t''}$ respectively.
4. The payment of agent p for question j is $S(r_p^j, r_q^j) - S(r_q^{t'}, r_q^{t''})$.

Here $S : [n] \times [n] \to \{0, 1\}$ is a scoring matrix which maps two signal reports (with at most n possible values) to a $0-1$ score. We make some adjustments to the CA mechanism described above. We defined the scoring matrix so that it takes as input two continuous forecasts instead of two discrete signals. In order to achieve this, we discretize the interval $[0, 1]$ into 10 bins and compute the delta matrix using reports from the good judgement platform Ungar et al. (2012). For the exact details about the scoring matrix, see the Supplementary material (subsection 7.3).

Finally, we adjust the payments so that the workers get paid 4 cents on average for each task and each day. We did so by collecting the reports from the good judgement platform and then normalizing both the CA score and Brier score so that in expectation both of them pay 4 cents. This implies that for each question and each task, the treatments SR, PP and SR+PPRank pays 4 cents on average. The treatment SR+PP uses both CA and Brier score. So, only for this treatment we normalized the scores so that on average both CA and Brier score pay 2 cents. This guarantees that each treatment provides the same payment in expectation for all the questions.

2.3 The Task

The main HIT starts with a tutorial of the payment method. We did not provide explicit formula for the payment, but provided an interface where the players can change their forecast and observe how their bonus payments change. When the workers accept the HIT, they go therough the tutorial corresponding to their treatments. Then they provide forecasts for the questions. The HIT was open for seven days and the workers could have come back to the platform anytime during the seven days, retake their tutorials if they wanted to, and update their forecasts.

3 User Engagement

Providing intermediate feedback, either in terms of money or in terms of rank, should motivate the workers to come back to the platform more often and update their predictions. This implies that the treatment SR should have the least updates and/or changes in forecast among the four treatments. We next verfiy this is indeed the case by comparing the four different treatments across three possible statistics that captures how users engage with the platform. Figure 2 plots the following three statistics and the corresponding 95% confidence intervals.

1. **Updates**: the total number of updates made by a user over the whole week on a question.
2. **Returns**: the frequency of returns of a user on the platform.

3. **Change**: the average amount of change made to a question by a user.

Note that the three statistics progressively capture finer details about the user's engagements with the platform. Updates indicate the total number of updates made by a user over the whole week and might be misleading if the users make most of their updates at the start of the experiment. Returns account for such events but fail to capture the magintude of changes made by a user on a forecast. Change, on the other hand, computes the exact amount of changes made by a user on forecasts for a question.

We observe that the treatments SR+PP and PP performs significantly better than SR for all three statistics (top row of figure 2). However, SR+PPRank performs significantly worse than SR for the number of daily returns. To understand this phenomenon, we next see what happens if we consider "good" users i.e. the top-75 most updating users on the final day of the prediction. [5] We see a similar pattern for treatments PP and SR+PP (bottom row of figure 2, but SR+PPRank performs significantly better than SR in terms of updates and change, and shows no siginifcant difference in terms of daily returns. Since providing monetary or non-monetary feedback increases users' engagement with the platform, we posit that such increase in engagement brings more information to the platform and should increase accuracy of the forecasts.

4 Average Final Score

We first compare the average score of the final forecasts under the four treatments. This is calculated by first computing the brier score of each forecast on the final day under each treatment and then averaging the scores. If a particular payment scheme incentivizes the users to provide higher quality forecasts, then all the forecasts under that treatment will also have higher Brier score and this should provide a higer average of final Brier scores.

We start with a simple null hypothesis: the distribution of the final score is the same under the four treatments. [6] To test this hypothesis, we run Kruskal-Wallis test Kruskal and Wallis (1952), a non-parametric method to check whether two or more groups of samples originate from the same distribution. This test rejects the null hypothesis with p-value 0.0075. However, it does not idetify which pairs of treatments are different. So we run a pairwise Wilcoxon rank sum test with the false discovery rate controlled by Benjamini and Hochberg (1995) and found that the treatment SR+PP is siginificantly different from the other three treatments.

Treatment	Mean Final Accuracy
SR	0.725 (0.713, 0.736)
PP	0.720 (0.709, 0.731)
SR+PPRank	0.726 (0.715, 0.737)
SR+PP	0.717 (0.706, 0.728)

Table 1: Mean Final Score and its 95% Confidence Interval. The highlighted entry indicates highest mean score. Statistical tests show that the distribution of SR+PPRank is significantly different than the other three treatments, but the differences among their means are not signficant.

Table 1 lists the average final score under the four treatments. Although, the treatment SR+PPRank has higher mean than the other treatments, there are not significant differecnes among the means of different treatments. So, we next consider average final score separately for dynamic and static questions and easy and hard questions. Table 2 shows the average final scores for (dynamic,static) split and (hard,easy) split of the questions. We repeated the same analysis as before, and found that the statistical test cannot reject the null (the distributions of scores are the same) for both static and easy questions. However, we do find that the distribution of the final scores for SR+PP is significantly different than SR+PPRank and SR under both dynamic and hard questions.

5 Accuracy Under Different Aggregators

We now see how different treatments perform when we use the same aggregator to produce an estimate of the event for the days the experiment was run. In particular, we consider two classes of aggregators. The first class (aggregators **1.a** through **1.d**) are based on the mean aggregator.

1.a: **Mean**: aggregator computes the sample mean.

1.b: **Weighted Mean**: aggregator computes weighted mean where the weights are proportional to the number of updates made by an individual on a particular question.

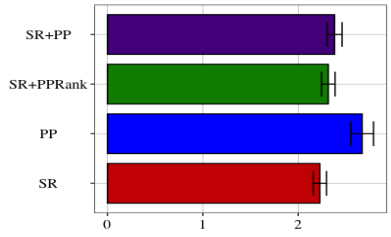
1.c: **Top-k + Weighted Mean**: aggregator computes weighted mean using the forecasts of the top k users in terms of the number of updates. [7]

1.d: **Top-k + Weighted Mean + Extremize**: After computing the weighted mean of the top-k users, the aggregator extremizes the forecast using the following formula Atanasov et al. (2016): $\bar{f}_e =$

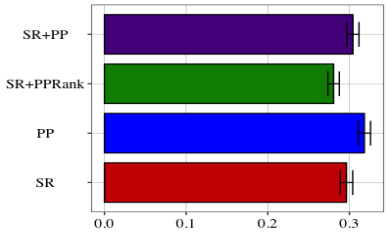
[5] We will later see that considering only the top-k users improves the accuracy.

[6] As an alternative, we could have considered the following: the final accuracy is the same under the four treatments. However, the data fails the normality test. So we cannot use a parametric test like one-way ANOVA test to reject the null hypothesis Casella and Berger (2002).

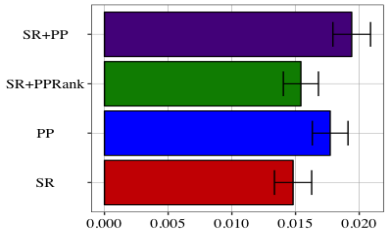
[7] We used $k = 75$ for our setting. The choice of k exhibits a bias-variance trade-off. Picking a small k improves accuracy but increases the variance. We saw that the aggregated forecast remained unchanged unless k was less than 90 (# users $>$ 200 for each treatment). We leave the choice of optimal k for future work.



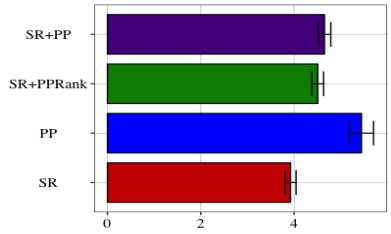
(a) Average Number of Updates over the Week



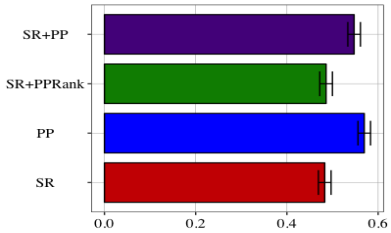
(b) Daily Returns



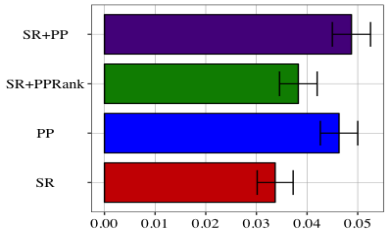
(c) Average Change in Forecast



(d) Average Number of Updates over the Week (Top 75 Users)



(e) Daily Returns (Top 75 Users)



(f) Average Change in Forecast (Top 75 Users)

Figure 2: The top row shows the three statistics (updates, returns, and change) for the four treatments. Treatments SR+PP and PP performs significantly better than SR for all three statistics, however, SR+PPRank performs significantly worse than SR for the number of daily returns. To understand this phenomenon, the bottom row shows the corresponding figures when we only consider "good" users i.e. top-75 most updating users on the final day of the prediction. We see a similar pattern for treatments PP and SR+PP, but SR+PPRank performs significantly better than SR in terms of updates and change, and shows no siginifcant difference in terms of daily returns.

.

Treatment	Dynamic	Static	Easy	Hard
SR	0.702 (0.683, 0.721)	0.747 (0.730, 0.765)	0.777 (0.763, 0.792)	0.641 (0.619, 0.664)
PP	0.696 (0.678, 0.715)	0.744 (0.726, 0.761)	0.769 (0.754, 0.783)	0.643 (0.622, 0.666)
SR+PPRank	0.703 (0.685, 0.720)	0.750 (0.733, 0.766)	0.772 (0.757, 0.786)	0.655 (0.635, 0.675)
SR+PP	0.690 (0.672, 0.707)	0.744 (0.728, 0.760)	0.761 (0.747, 0.776)	0.647 (0.627, 0.666)

Table 2: Mean Final score and its 95% confidence intervals for (dynamic,static) split and (hard,easy) split of the questions. Statistical tests show that the distribution of the final scores for SR+PP is significantly different than SR+PPRank and SR under both dynamic and hard questions. The highlighted entries indicate the best treatment in terms of mean final score. We see that the treatment SR+PPRank performs well for many cases, but we did not find any statistically significant difference betweeen the means for different partitions of the questions.

$\bar{f}^a/(\bar{f}^a + (1-\bar{f})^a)$. Atanasov et al. (2016) found that the optimal value of a was 2 for the good judgement project and we use the same for our setting.

The second set of aggregators (**2.a** to **2.d**) are based on the logit aggregation rule Satopää et al. (2014) and are similar to the aggregators **1.a** to **1.d**.

2.a: **Logit**: aggregator computes the average logit $\bar{y} = 1/n \sum_{i=1}^{n} \log \frac{f_i}{1-f_i}$ and then computes the inverse-logit for the final forecast, $\bar{f} = \frac{\exp(\bar{y})}{1+\exp(\bar{y})}$.

2.b: **Weighted Logit**: aggregator computes weighted logit and then computes the inverse-logit as before.

2.c: **Top-k + Weighted Logit**: computes weighted logit over the top-75 users according to their updates.

2.d: **Top-k + Weighted Logit + Extremize**: aggregator extremizes the forecast provided by **2.c**.

Table 3 lists the performances of the four treatments. We see that **1.d** (resp. **2.d**) gives the highest accuracy among the mean (resp. logit) based aggregators. Therefore, we focus on comparing the performance of the four treatments for these two aggregators. We computed the 95% confidence intervals using bootstrap, but found no significant difference in the final accuracy averaged over all the questions. However, we again see that the treatment SR+PPRank performs best for many types of questions (table 4).

Section 7.4 in the supplementary material plots the performance of all the aggregators over the entire experiment. Here we summarize our main findings.

Treatment	**1.a**	**1.b**	**1.c**	**1.d**	**2.a**	**2.b**	**2.c**	**2.d**
SR	0.804	0.810	0.819	0.847	0.842	0.859	0.876	0.872
PP	0.799	0.797	0.794	0.817	0.826	0.828	0.825	0.834
SR+PPRank	0.805	0.812	0.822	0.856	0.836	0.857	0.882	0.891
SR+PP	0.790	0.794	0.798	0.823	0.818	0.829	0.834	0.851

Table 3: Final Accuracy under Different Aggregators. We see that **1.d** (resp. **2.d**) gives the highest accuracy among the mean (resp. logit) based aggregators. The blue entries highlight the treatment with the highest final accuracy under a given aggregator. We see that the treatment SR+PPRank performs best under many aggregators. We also computed 95% confidence intervals of the final accuracy using bootstrap but omit them from the table due to space constraint (see section 7.4 in the supplementary material for details).

Treatment	Dynamic	Static	Easy	Hard
SR	0.834	0.861	0.929	0.718
PP	0.794	0.841	0.876	0.724
SR+PPRank	0.839	0.874	0.912	0.769
SR+PP	0.779	0.867	0.893	0.714

Aggregator **1.d**

Treatment	Dynamic	Static	Easy	Hard
SR	0.854	0.890	0.991	0.684
PP	0.840	0.829	0.921	0.698
SR+PPRank	0.909	0.872	0.963	0.778
SR+PP	0.784	0.919	0.950	0.697

Aggregator **2.d**

Table 4: Final accuracy under aggregators **1.d** and **2.d** for two different partitions – (dynamic, static) and (easy, hard). We see that the treatment SR+PPRank peforms best for many types of questions. We also computed the 95% confidence intervals using bootstrap (see section 7.4 in the supplementary material for details).

1. For the static questions, there are no significant differences among the performances of the four treatments.
2. For both dynamic and easy questions, treatments SR and SR+PPRank work best, and signifcantly outperform the other two treatments for some aggregators.
3. For hard questions, treatment SR+PPRank provides highest final accuracy, but there are no significant differences among the four treatments.

6 Conclusion

We started this paper with two main questions, whether providing intermediate feedback (monetary or non-monetary) increases users' engagement to the platform and whether they boost the accuracy of the forecasts. The answer to the first question is positive as we saw that the treatments providing monetary and/or non-monetary feedback perform significantly better than the treatment SR for various statistics capturing user engagement to the platform. On the other hand, the answer to our second question is more subtle. Whether an increase in updates boosts the overall performance depends on the particular type of aggregator, and also on the kind of questions considered. We considered eight aggregators in total and found that the treatments SR and SR+PPRank perform best for both dynamic and easy questions, and signifcantly outperform the other two treatments for some aggregators. Since SR+PPRank improves user engagement compared to SR, this leads us to recommend providing non-monetary feedback based on peer prediction scores for improving the performance of various forecasting platforms.

We observed that providing monetary feedback (treatments SR+PP and PP) signifcantly improves user engagement compared to SR. However, for easy questions, they hurt the accuracy of the forecast under some aggregators. We believe this is because providing monetary feedback creates wrong incentives for easy questions, which require very few updates in forecasts over the duration of the experiment. It will be interesting to further investigate the nature of such interim monetary incentives on forecasting in future.

We found no significant difference in final accuracy between the treatments SR+PPRank and SR, even though SR+PPRank improves user engagement significantly than SR. We think that the duration of our experiment was too short to differentiate these two treatments and SR+PPRank will indeed be the best treatment for longer term forecasting events. However, we would like to note that it is quite challenging to run such an experiment for a long term (say several months) on the Amazon Mechanical Turk, since the users tend to communicate among each other Yin et al. (2016) and this defeats the whole purpose of running different treatments in parallel. One major direction for future work is to run such an experiment in a properly controlled setting.

References

Atanasov, P.; Rescober, P.; Stone, E.; Swift, S. A.; Servan-Schreiber, E.; Tetlock, P.; Ungar, L.; and Mellers, B. 2016. Distilling the wisdom of crowds:

Prediction markets vs. prediction polls. *Management science* 63(3):691–706.

Benjamini, Y., and Hochberg, Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* 57(1):289–300.

Brier, G. W. 1950. Verification of forecasts expressed in terms of probability. *Monthey Weather Review* 78(1):1–3.

Casella, G., and Berger, R. L. 2002. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA.

Dasgupta, A., and Ghosh, A. 2013. Crowdsourced judgement elicitation with endogenous proficiency. In *Proceedings of the 22nd international conference on World Wide Web*, 319–330. ACM.

Faltings, B., and Radanovic, G. 2017. Game theory for data science: eliciting truthful information. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 11(2):1–151.

Gao, X. A.; Mao, A.; Chen, Y.; and Adams, R. P. 2014. Trick or treat: putting peer prediction to the test. In *Proceedings of the fifteenth ACM conference on Economics and computation*, 507–524. ACM.

Gao, A.; Wright, J. R.; and Leyton-Brown, K. 2016. Incentivizing Evaluation via Limited Access to Ground Truth: Peer-Prediction Makes Things Worse. *EC 2016 Workshop on Algorithmic Game Theory and Data Science*.

Garcin, F., and Faltings, B. 2014. Swissnoise: Online polls with game-theoretic incentives. In *AAAI*, 2972–2977.

Gneiting, T., and Raftery, A. E. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* 102(477):359–378.

Goel, N., and Faltings, B. 2019. Deep bayesian trust: A dominant and fair incentive mechanism for crowd. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1996–2003.

Hanson, R. 2012. Logarithmic markets coring rules for modular combinatorial information aggregation. *The Journal of Prediction Markets* 1(1):3–15.

IARPA. 2019. Hybrid Forecasting Competition (HFC).

Jurca, R., and Faltings, B. 2009. Mechanisms for making crowds truthful. *Journal of Artificial Intelligence Research* 34:209–253.

Kong, Y.; Ligett, K.; and Schoenebeck, G. 2016. Putting peer prediction under the micro (economic) scope and making truth-telling focal. In *International Conference on Web and Internet Economics*, 251–264. Springer.

Kruskal, W. H., and Wallis, W. A. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association* 47(260):583–621.

Mandal, D.; Leifer, M.; Parkes, D. C.; Pickard, G.; and Shnayder, V. 2016. Peer Prediction with Heterogeneous Tasks. *NIPS 2016 Workshop on Crowdsourcing and Machine Learning*.

Miller, N.; Resnick, P.; and Zeckhauser, R. 2005. Eliciting informative feedback: The peer-prediction method. *Management Science* 51(9):1359–1373.

Prelec, D. 2004. A bayesian truth serum for subjective data. *science* 306(5695):462–466.

Radanovic, G., and Faltings, B. 2015. Incentive schemes for participatory sensing. In *Proc. Int. Conf. on Autonomous Agents and Multiagent Systems, AAMAS*, 1081–1089.

Radanovic, G.; Faltings, B.; and Jurca, R. 2016. Incentives for effort in crowdsourcing using the peer truth serum. *ACM Transactions on Intelligent Systems and Technology (TIST)* 7(4):48.

Rigol, N., and Roth, B. 2017. Paying for the truth: The efficacy of peer prediction in the field. Technical report, Working Paper.

Satopää, V. A.; Baron, J.; Foster, D. P.; Mellers, B. A.; Tetlock, P. E.; and Ungar, L. H. 2014. Combining multiple probability predictions using a simple logit model. *International Journal of Forecasting* 30(2):344–356.

Shaw, A. D.; Horton, J. J.; and Chen, D. L. 2011. Designing incentives for inexpert human raters. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, 275–284. ACM.

Shnayder, V.; Agarwal, A.; Frongillo, R.; and Parkes, D. C. 2016. Informed truthfulness in multi-task peer prediction. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, 179–196. ACM.

Ungar, L.; Mellors, B.; Satopää, V.; Baron, J.; Tetlock, P.; Ramos, J.; and Swift, S. 2012. The good judgment project: A large scale test. Technical report, AAAI Technical Report.

von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *Proc. SIGCHI Conf. on Human Factors in Computing Systems (CHI'04)*, 319–326.

Witkowski, J., and Parkes, D. C. 2012. A Robust Bayesian Truth Serum for Small Populations. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Yin, M.; Gray, M. L.; Suri, S.; and Vaughan, J. W. 2016. The communication network within the crowd. In *Proceedings of the 25th International Conference on World Wide Web*, 1293–1303. International World Wide Web Conferences Steering Committee.

7 Supplementary Material

7.1 List of Questions

Id	*Question*	*Dynamic*	*Hard*	*Outcome*
1	Will D.C. United win the D.C. United vs. FC Dallas soccer game (Major League Soccer) on Sat Oct. 13th?	N	N	Y
2	Will the Chicago Bears win the Chicago Bears vs Miami Dolphins game on Sun Oct. 14th?	N	Y	N
3	Will the Seattle Seahawks score more than 60 points in total in the games Los Angeles Rams vs Seattle Seahawks on Sun. Oct 7th and the game Seattle Seahawks vs Oakland Raiders on Sun. Oct 14th?	Y	N	N
4	Will Italy win against Poland in the Uefa Nations league match on Sun. Oct 14th?	N	N	Y
5	Will the Houston Texans score more points in their games against Dallas Cowboys on Sun. Oct 7th than their game against Buffalo Bills on Sun. Oct 14th?	Y	N	N
6	Will Orlando City score more goals in their game against the FC Dallas on Sat Oct 6th than their game against the New England on Sat Oct 13th?	Y	Y	N
7	Will the price of Bitcoin in USD on Sat Oct 13th (EST time zone) be, at any point of the day, above 6500? (resource: https://xe.com/currencycharts/?from=XBT&to=USD)?	Y	Y	N
8	Will the end-of-day (EST time zone) closing value for the British pound against the US dollar drop below $1.32 on Mon Oct. 15th?	Y	Y	Y
9	Will Facebook's stock price quote on NASDAQ on Sat Oct. 13th go above $160? (resource: https://www.nasdaq.com/symbol/fb)	Y	N	N
10	Will one of Alphabet Inc., Facebook, Amazon, or Microsoft make an announcement on Sat Oct. 13th (EST time zone) about a security breach?	N	N	N
11	Will Theresa May propose a new Brexit plan on Mon Oct. 15th (BST time zone)?	N	N	N
12	Will Donald Trump fire a member of the White House staff on Mon Oct. 15th (EST time zone)?	N	N	N
13	Will the official YouTube video of the Taylor Swift song 'Delicate' reach 300,000,000 views before Sat Oct. 13th midnight (EST time zone)?	Y	Y	N
14	Will 'Crazy Rich Asians' have a 'Rotten Tomatoes' score (in tomatometer) above 90% on Sat Oct 13th at noon (EST time zone)?	Y	N	Y
15	Will Elon Musk tweet more than 5 times on Sat Oct. 13th (EST time zone)?	N	Y	N
16	Will Trevor Noah (a comedian) be more popular than John Oliver (a comedian) on Google Trends (https://trends.google.com/trends/) for settings (United States, All Categories, Web Search) on Sat Oct. 13th at noon (EST time zone)?	Y	Y	Y
17	Will Ryan Gosling have more than 2.4 Million followers on Twitter by Sat Oct. 13th midnight (EST time zone)?	N	N	N
18	Will more than five patents containing 'Blockchain' in the title be published on Sat Oct. 13th (EST time zone) in the online search repository of the US patent system? (resource: http://appft.uspto.gov/netahtml/PTO/search-adv.html)?	N	N	N

7.2 Hard versus Easy Split

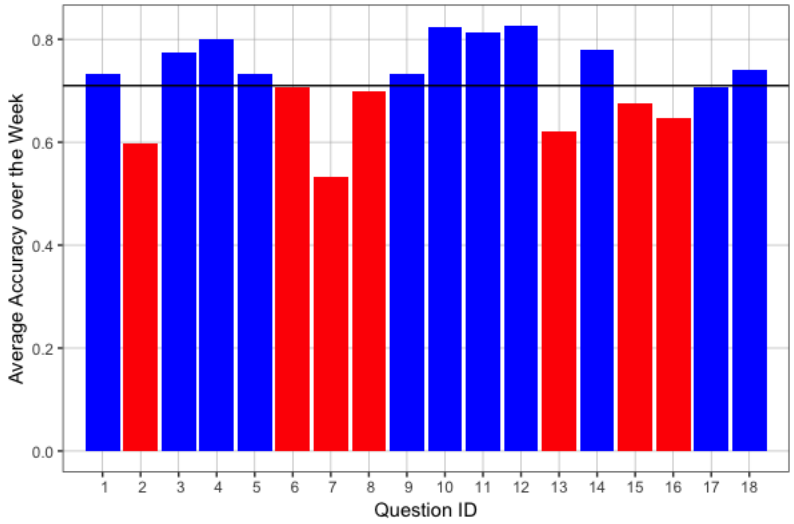
7.3 Scoring Matrix

CA uses the following scoring matrix :

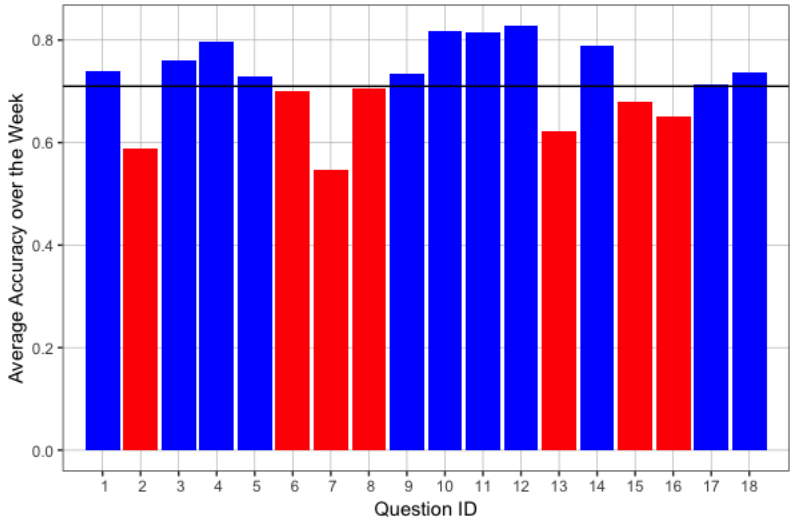
$$S(i,j) = \text{Sign}(P(i,j) - P(i)P(j)) = \text{Sign}(\Delta(i,j))$$

Here $P(i,j)$ is the joint probability of observing the signal pairs i and j and $\Delta(i,j)$ measures the correlation between the signals i and j. Suppose the signals are positively correlated i.e. $\Delta(i,i) > 0$ and $\Delta(i,j) \leqslant 0$ for $i \neq j$. In that case, CA rewards for an agreement on the same question and punishes for an agreement on two separate questions.

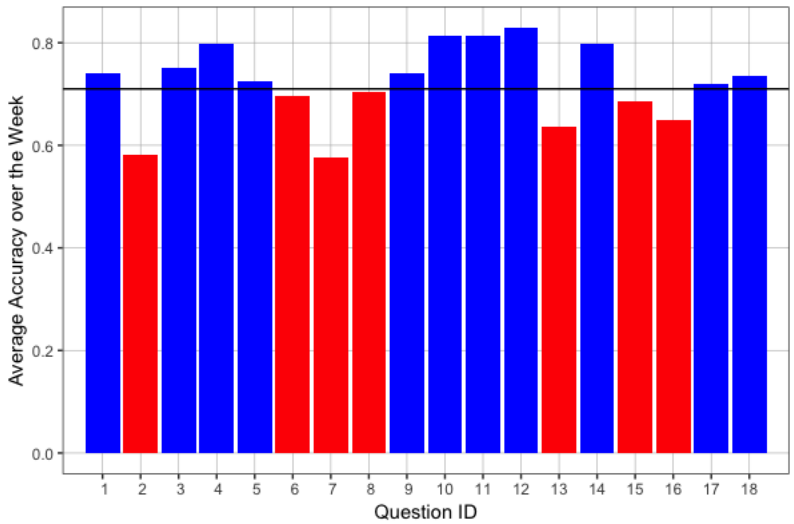
We make two adjustments to the CA mechanism described above.



(a) Average Accuracy on October 8



(b) Average Accuracy over the Whole Week



(c) Average Accuracy on Oct 13

Figure 3: The average accuracy of the predictions for the 18 questions. We computed the average accuracy for the predictions made on the first day, last day and over the whole week. Questions 2,7,13,15, and 16 have average accuracy consistenly below 0.71 and we label these questions as "hard" and the remaining questions as "easy".

1. On a given day, CA awards a given response to a question by pairing it with a response from another user. This might produce a reward with high variance, so we compute the reward by averaging the scores obtained by pairing with 100 users.
2. We defined the scoring matrix so that it takes as input two continuous forecasts instead of two discrete signals. In order to achieve this, we discretize the interval $[0, 1]$ into 10 bins and compute the delta matrix using reports from the good judgement platform Ungar et al. (2012).Figure 4 displays the scoring matrix for two reports.

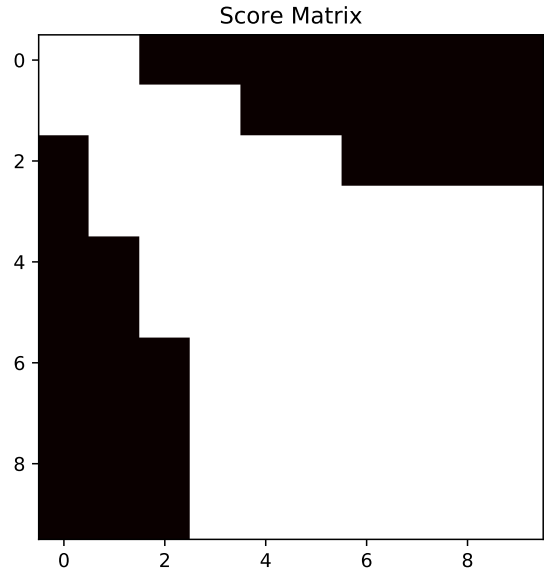
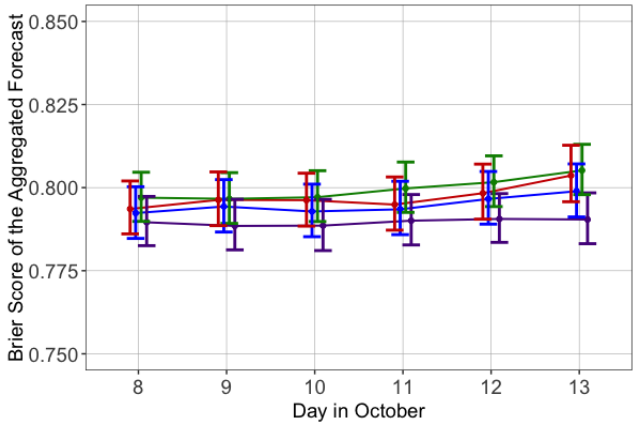
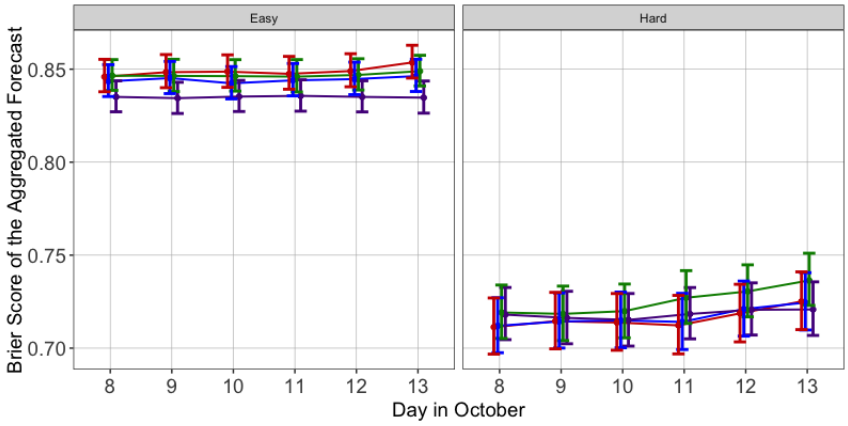


Figure 4: Score Matrix

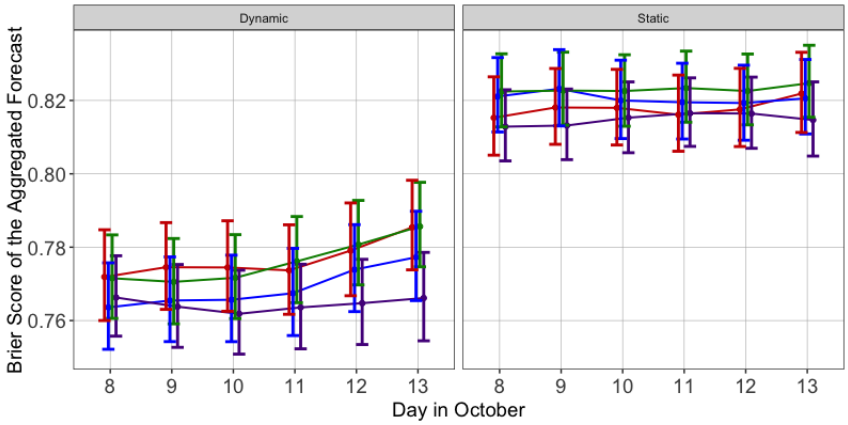
7.4 Performance of Different Aggregators



(a) All questions combined

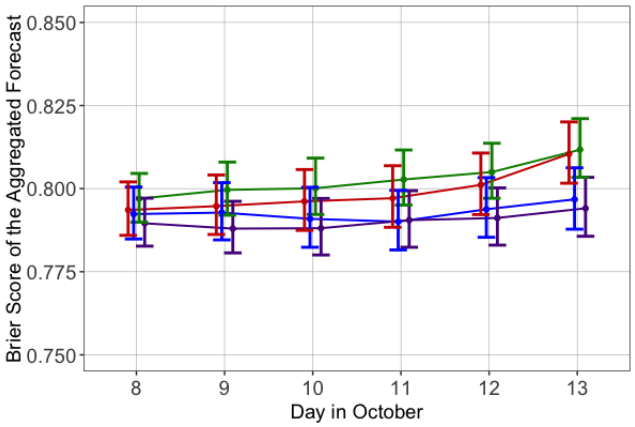


(b) Easy versus Hard Questions

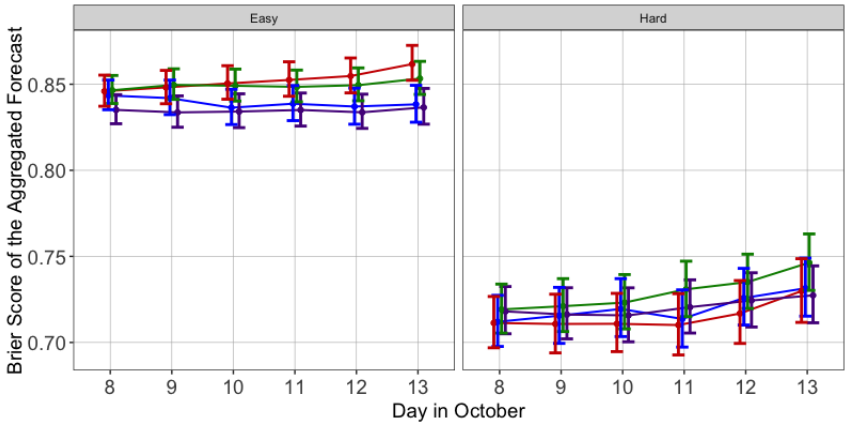


(c) Dynamic versus Static Questions

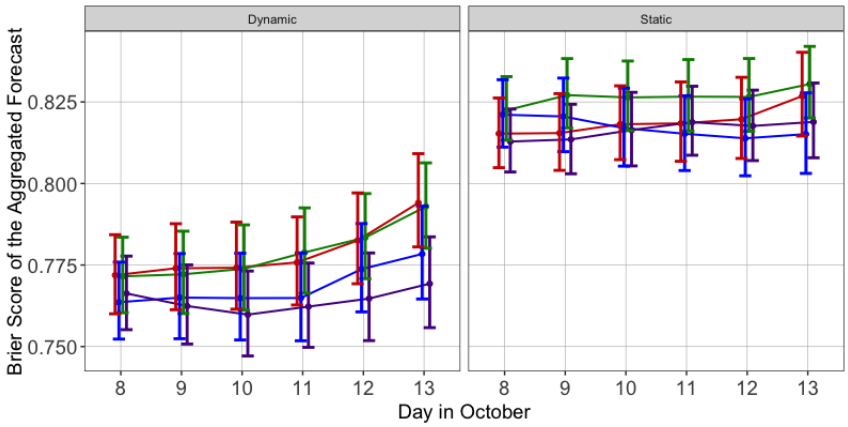
Figure 5: Aggregator 1.a: Mean



(a) All questions combined

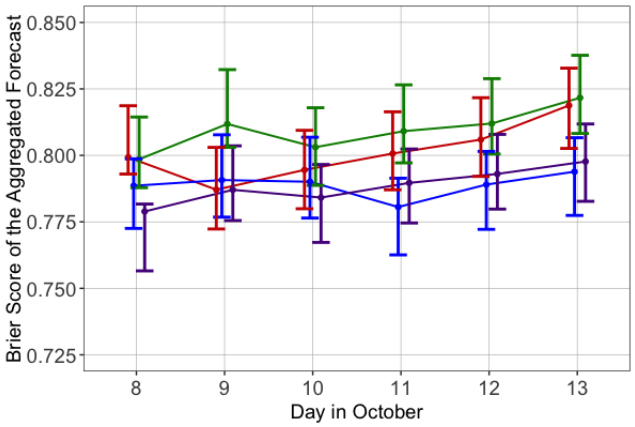


(b) Easy versus Hard Questions

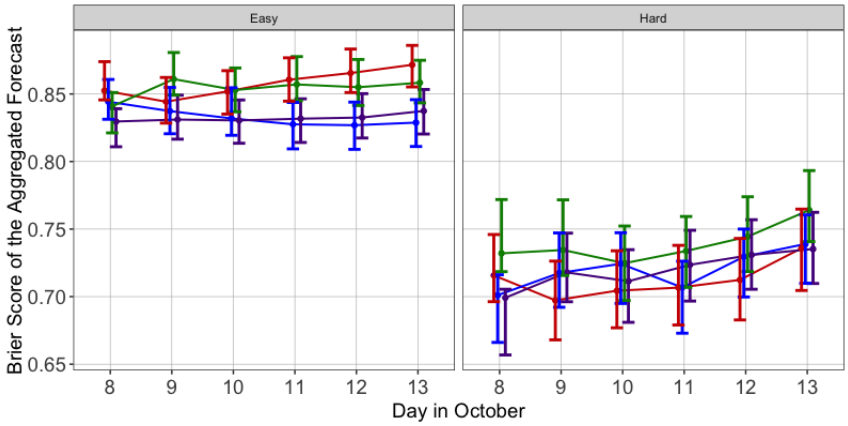


(c) Dynamic versus Static Questions

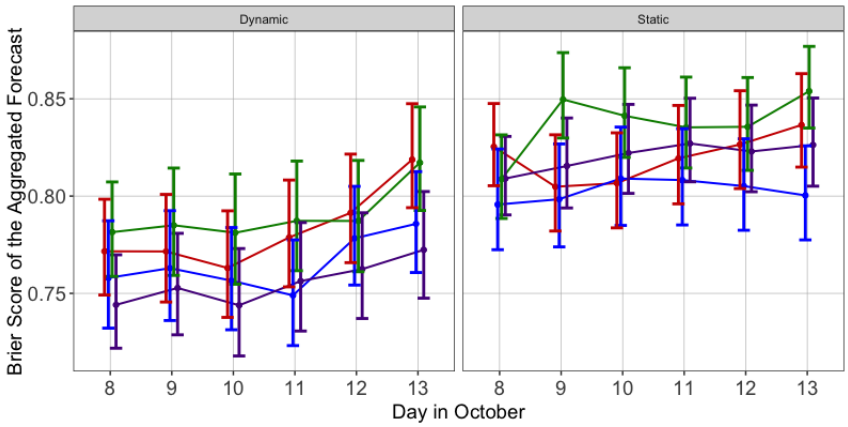
Figure 6: Aggregator 1.b: Weighted Mean



(a) All questions combined

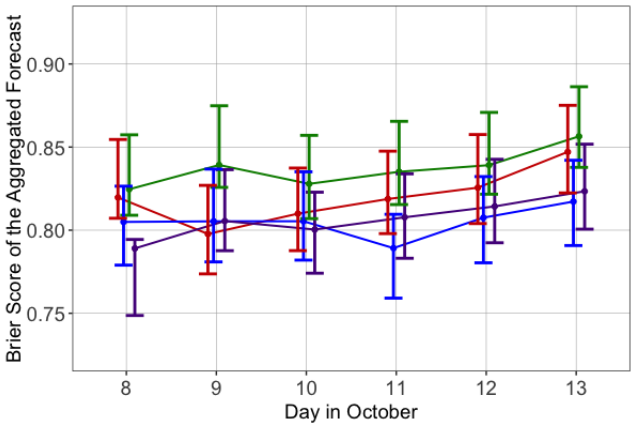


(b) Easy versus Hard Questions

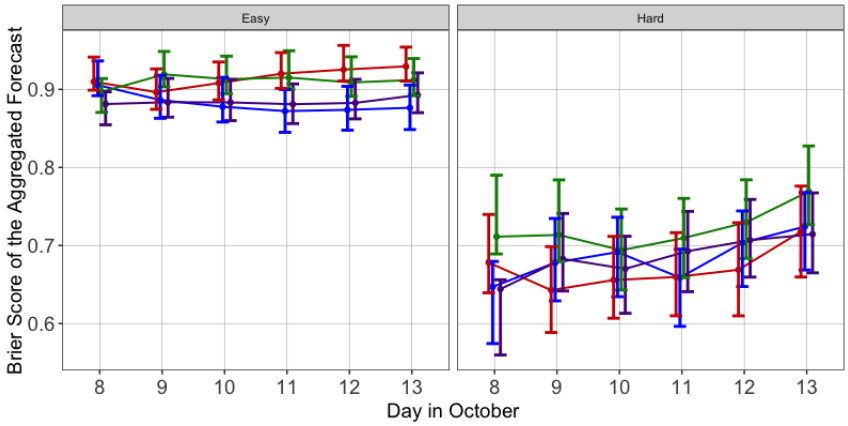


(c) Dynamic versus Static Questions

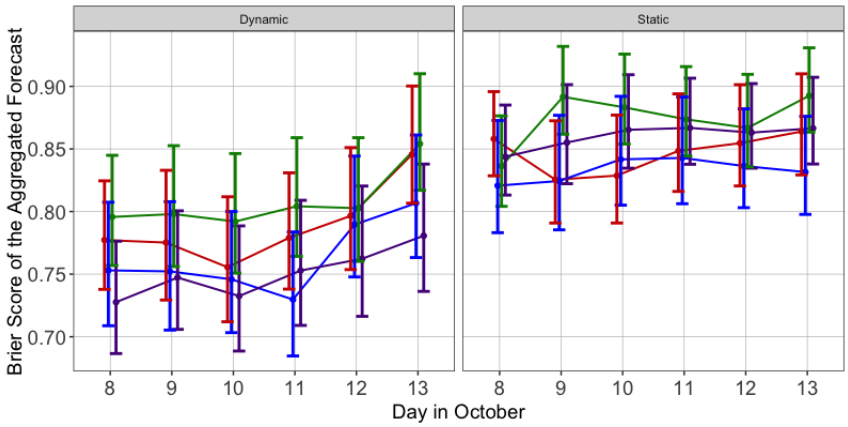
Figure 7: Aggregator 1.c: Top-k + Weighted Mean



(a) All questions combined

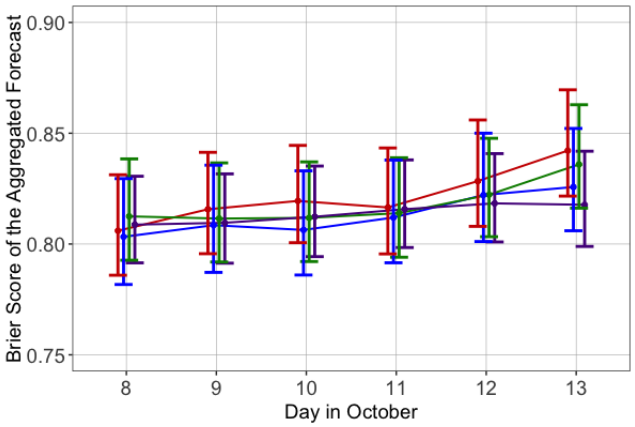


(b) Easy versus Hard Questions

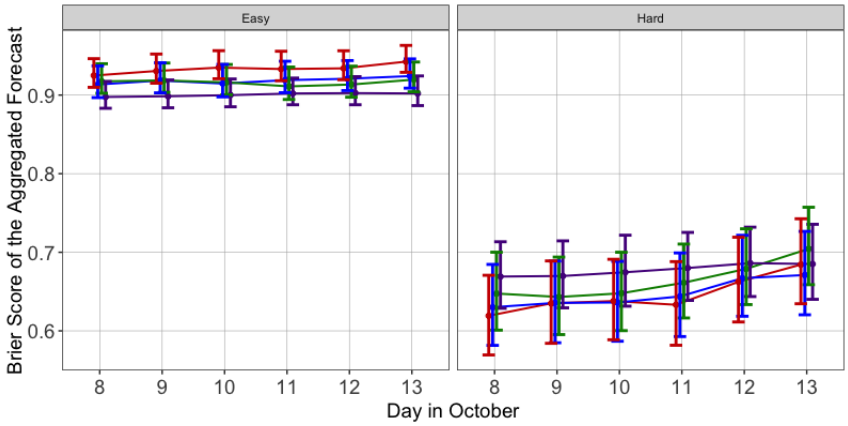


(c) Dynamic versus Static Questions

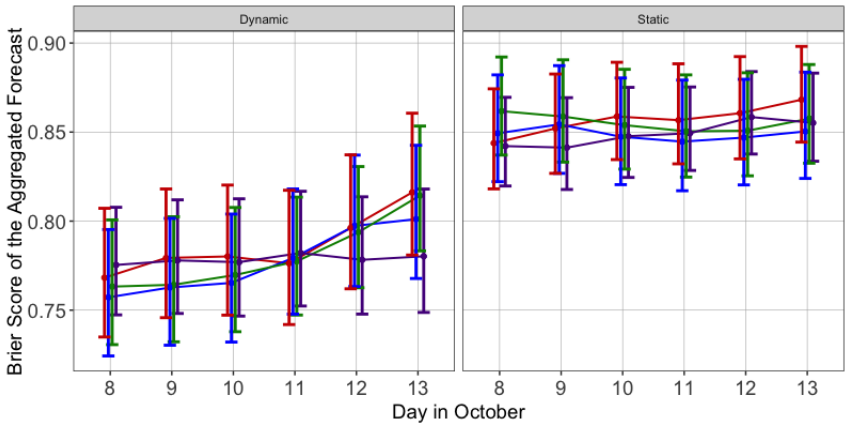
Figure 8: Aggregator 1.d: Top-k + Weighted Mean + Extremize



(a) All questions combined

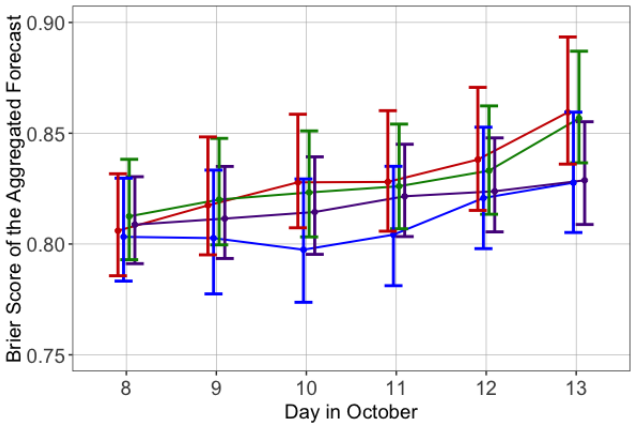


(b) Easy versus Hard Questions

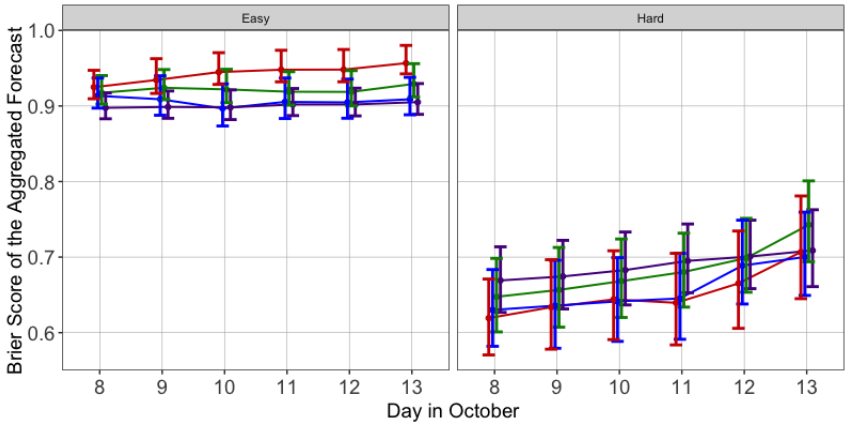


(c) Dynamic versus Static Questions

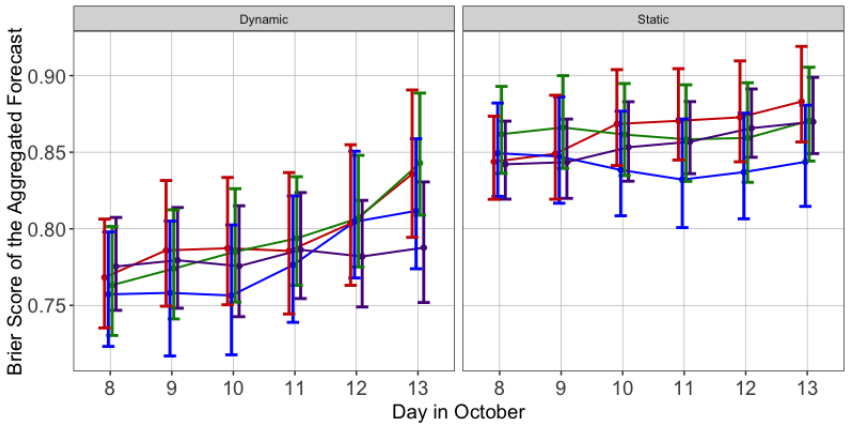
Figure 9: Aggregator 2.a: Logit



(a) All questions combined

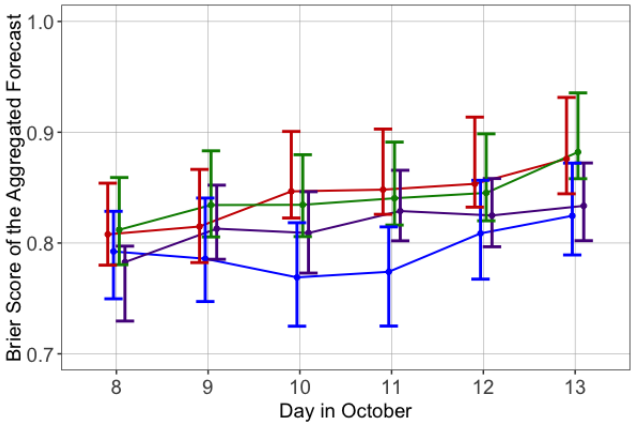


(b) Easy versus Hard Questions

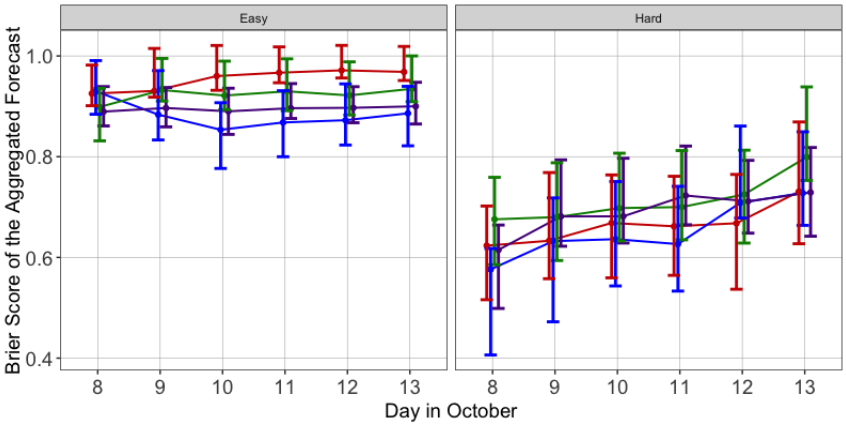


(c) Dynamic versus Static Questions

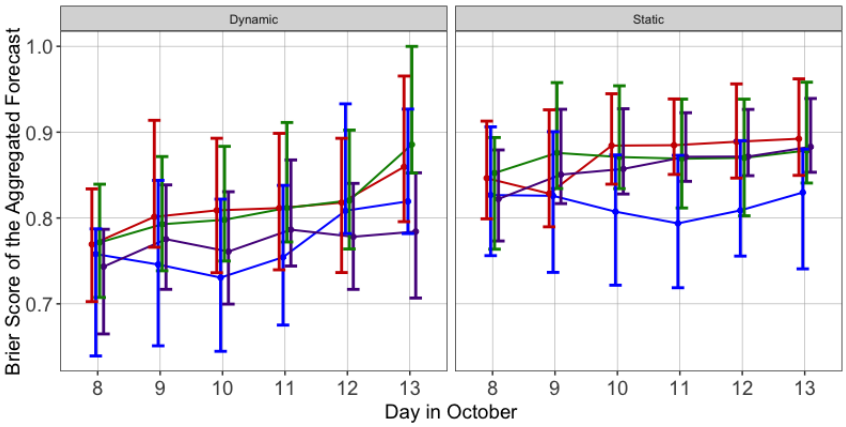
Figure 10: Aggregator 2.b: Weighted Logit



(a) All questions combined

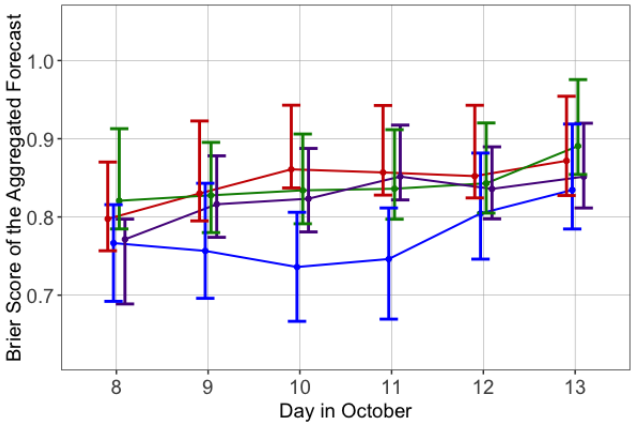


(b) Easy versus Hard Questions

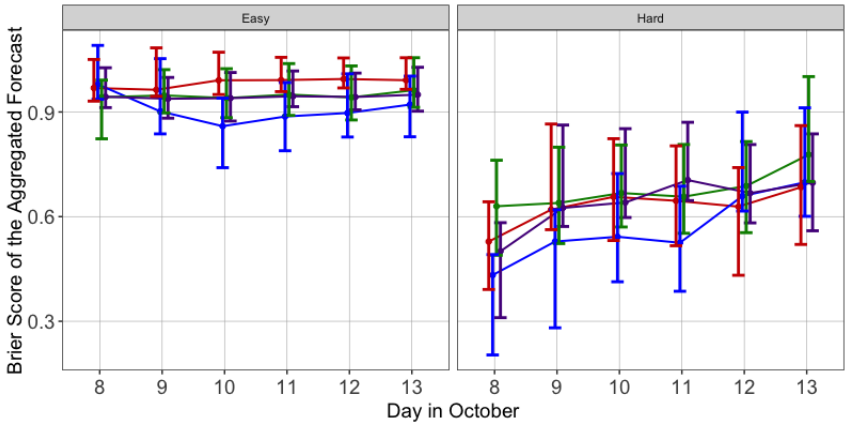


(c) Dynamic versus Static Questions

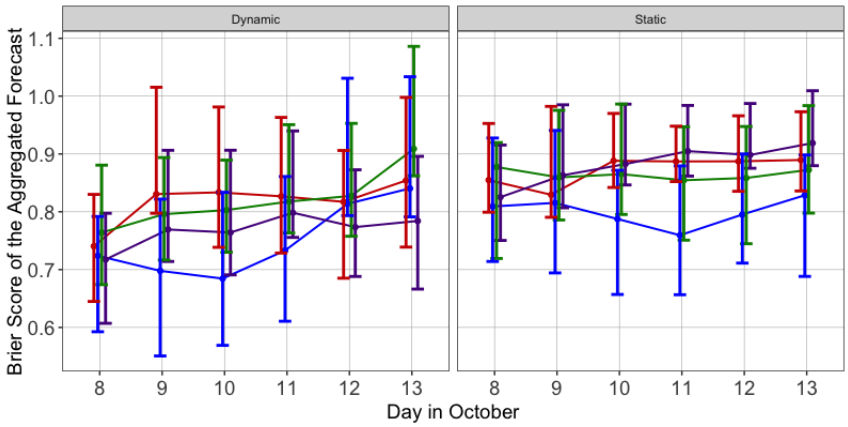
Figure 11: Aggregator 2.c: Top-k + Weighted Logit



(a) All questions combined



(b) Easy versus Hard Questions



(c) Dynamic versus Static Questions

Figure 12: Aggregator 2.d: Top-k + Weighted Logit + Extremize