A Crowdsourcing Triage Algorithm for Geopolitical Event Forecasting

Mohammad Rostami
University of Pennsylvania
Philadelphia, PA
mrostami@seas.upenn.edu

David Huber
HRL Laboratories, LLC
Malibu, CA
djhuber@hrl.com

Tsai-Ching Lu
HRL Laboratories, LLC
Malibu, CA
tlu@hrl.com

ABSTRACT

Predicting the outcome of geopolitical events is of huge importance to many organizations, as these forecasts may be used to make consequential decisions. Prediction polling is a common method used in crowdsourcing platforms for geopolitical forecasting, where a group of non-expert participants are asked to predict the outcome of a geopolitical event and the collected responses are aggregated to generate a forecast. It has been demonstrated that forecasts by such a crowd can be more accurate than the forecasts of experts. However, geopolitical prediction polling is challenging because participants are highly heterogeneous and diverse in terms of their skills and background knowledge and human resources are often limited. As a result, it is crucial to refer each question to the subset of participants that possess suitable skills to answer it, such that individual efforts are not wasted. In this paper, we propose an algorithm based on multitask learning to learn the skills of participants of a forecasting platform by using their performance history. The learned model then can be used to recommend suitable questions to forecasters. Our experimental results demonstrate that the prediction accuracy can be increased based on the proposed algorithm as opposed to when questions have been randomly assigned.

KEYWORDS

Geopolitical forecasting; Multitask learning; Biconvex optimization; Markov Chain Monte Carlo; MetropolisâĂŞHastings algorithm

ACM Reference Format:
Mohammad Rostami, David Huber, and Tsai-Ching Lu. 2018. A Crowdsourcing Triage Algorithm for Geopolitical Event Forecasting. In *Twelfth ACM Conference on Recommender Systems (RecSys '18), October 2–7, 2018, Vancouver, BC, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3240323.3240385

1 INTRODUCTION

To improve decision and policy making procedures, governments and other organizations continue to invest heavily in methods of predicting the outcome of various geopolitical events. It is estimated that market research and public opinion polling is a 17 billion dollar, yet growing, industry in the United States alone. To make forecasts, these organizations typically employ a polling method called "prediction polling", which asks a group of participants to predict the *outcome* of an event, rather merely expressing their personal tendency, such as with opinion polling; their responses are then aggregated to make a forecast [1]. The rationale behind predition polling is the idea that people with diverse backgrounds and knowledge can collectively predict future events more accurately by addressing the forecasting problem from different points of view [12]. It has been demonstrated that this strategy can allow a group of novices to outperform individual trained experts with access to classified information [13]. For this reason, prediction polling has gained much attention among researchers, which has given rise to the emergence of crowdsourcing platforms for a wide range of geopolitical applications, including intelligence analysis [4, 7] and policy debates [14].

Prediction polling is challenging because there exists a great deal of heterogeneity among the participants in terms of their skills and background knowledge and because of the resource management involved. As the complexity and required research effort to answer a question increases, the number of questions that a given forecaster can address must generally decrease. To overcome these challenges, one should first identify the most accurate forecasters by monitoring their performance during initial forecasting tasks and then leverage their skills to improve the prediction accuracy in future tasks. A long term study has demonstrated that this strategy can dramatically increase the accuracy of forecasts against a controlled group of randomly selected forecasters [8]. Building upon this result and inspired by the crowdsourcing literature [2], our goal is to develop a new scheme based on transfer and multitask learning [3] that identifies skilled forecasters and their skill level for particular forecasting topics. The scheme then can be used to triage questions to those participants who have the most suitable skills to answer those questions correctly, while also addressing the scalability and data sparsity issues that arise from a limited participant pool.

2 RELATED WORK

Crowdsourced geopolitical forecasting consists of two major steps: data collection and aggregation. The major challenge of data collection is to recommend questions to suitable and reliable participants. However, relying on simple techniques such as majority voting for aggregating the collected responses is highly inefficient because highly skilled and motivated participants are likely to be a minority in the crowd [2]. Moreover, non-agreement among the forecasters can be ascribed to their diverse information and background knowledge [12], rather than as additive noise, and therefore using statistical mean discards this diversity of skills among the participants [12]. Thus, one must take into the account both the skills of participants and the required skills to answer a question in the data

RecSys '18, October 2–7, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5901-6/18/10…$15.00
https://doi.org/10.1145/3240323.3240385

collection and aggregation steps. The simplest approach is to survey the participants about their skills and interests and incorporate this into the platform design. However, self-reported skills can be noisy, and potentially uncorrelated (or even negatively correlated) with a given task (e.g., the so-called Dunning-Kruger effect). A better approach is to learn these parameters through their activity as forecasters.

Participant heterogeneity has been studied in the crowdsourcing literature, where the core idea is to learn the expertise of participants given their performance history on past questions or tasks [2, 9, 16]. Whitehill et al. [16] were among the first to propose a probabilistic framework to infer both the expertise of each crowdsourcing participant and the difficulty of each question simultaneously in an iterative procedure. They parametrized the "probability of correct answer to a question by a user" as a function of the "participant expertise" and "question difficulty" and then used the expectation maximization (EM) algorithm to search for probabilistically optimal values of these parameters and the maximum likelihood (ML) estimate of the answers given the performance history. Bachrach et al. [2] proposed a similar approach, but they employed a different parametrization scheme and used message passing to learn the proposed graphical model. Both of the above and other similar approaches suffer from the problem of data sparsity, meaning that often few participants might answer to a particular question and each participant might answer only a few questions [9]. This problem is more challenging for geopolitical forecasting than for common crowdsourcing questions, such as the binary classification of an image, due to the complexity of the questions (e.g., "Will the U.S. carry out a preemptive strike against North Korea if 2018 bilateral talks fail?") and increased cognitive load of the task, which necessarily limits the number of questions that a single forecaster will be able to answer.

The challenge of data sparsity can be addressed within the transfer learning (TL) [10] and multitask learning (MTL) [3] frameworks. The core idea is that learned knowledge can be transferred across multiple related tasks and by exploiting commonalities among tasks, they can be learned more efficiently using less data than when learned in isolation. Knowledge transfer is unilateral in TL from (a) source task(s) to (a) target task(s), whereas in MTL, knowledge transfer is bilateral and the goal is to increase performance across all tasks. Mo et al. [9] benefited from transfer learning to improve performance of crowdsourcing. Their idea is built upon the framework proposed in [16] and their novelty is to group the crowdsourcing questions into tasks, where questions in each task share similarities, such as user ratings for various makeup products. They learn task-specific factors to model task differences and the users' skills for each task. Their idea is to use the learned knowledge from source tasks to improve performance in a given target task. They use a hierarchical Bayesian model and then model parameters are estimated using Markov Chain Monte Carlo (MCMC) sampling.

Building upon the same rationale, we investigate crowdsourced geopolitical forecasting in a multitask learning setting. We hypothesize that the performance of a participant depends on how their skills match those required to answer the questions of a task. We assume that for each participant, the skills for a particular task can be modeled by a vector, and in order to benefit from MTL to learn the task-specific skills for a participant, we further hypothesize that these vectors lie in a low-dimensional subspace [11]. The intuition behind our formulation is that the task-specific skills of a participant for a particular task can be represented as a combination of basic skills, such as math or knowledge in politics, which are represented by the columns of a dictionary.

3 PROPOSED FRAMEWORK

Consider the tasks $\{\mathcal{T}^{(t)}, 1 \leq t \leq T\}$ with each having the question set $\boldsymbol{Q}^{(t)} = \{Q_i^{(t)}\}_{i=1}^{n_t}$. The questions in each task are about future events and are related, meaning that conditioned on possessing a specific required background knowledge/information and a set of skills, a forecaster can predict the outcome of the corresponding future event in the task with high probability. The corresponding answer to each question is denoted by $A_i^{(t)} \in \mathcal{A}_i^{(t)}$, where $\mathcal{A}_i^{(t)}$ is a discrete set of multiple choices, e.g. $a, b, \cdots$ or YES/NO. There also exists a pool of forecasters $\{\mathcal{F}_j\}_{j=1}^F$ who might respond to these questions. When a question is referred to a given forecaster, he or she then potentially gives a response to the question as their prediction; we denote such a response as $R_{ij}^{(t)}$. It is likely that any given forecaster will respond to only a subset of the questions to which they are assigned, and also likely that others may respond to questions that were not recommended/triaged to them. Therefore, let $\mathcal{R} = \{R_{ij}^{(t)} | \forall (i,j,t), \mathcal{F}_j \quad \text{has responded to} \quad Q_i^{(t)}\}$ denote the set of all collected responses. The answers to the questions are not known when the predictions are collected; therefore, the goal is to predict the (future) answer of each question given the set of responses $\{R_{ij}^{(t)}\}$ for t by aggregating the collected responses from the crowd.

Following the previous discussion, we assume that $\boldsymbol{d}^{(t)} \in \mathbb{R}_+^d$ characterizes the task-specific required skills for each task, which are determined by the platform designer. Intuitively, each element of this vector can specify the required skill to answer questions of a task, such as knowledge of probability theory or political insight. Also, let $\boldsymbol{c}_j \in \mathbb{R}_+^d$ model the skills of each forecaster to handle tasks, which is drawn from a Gaussian distribution $\boldsymbol{c}_j \sim \mathcal{N}(\mu_f, \Sigma_f)$. The mean and the covariance of this distribution can be approximated from empirical studies. We propose that the ability of a particular participant to answer questions of a particular task, denoted by $s_j^{(t)} \in \mathbb{R}_+$ is drawn from the Gaussian distribution $s_j^{(t)} \sim \mathcal{N}(\langle \boldsymbol{d}^{(t)}, \boldsymbol{c}_j \rangle, \sigma_s)$, where $\langle \cdot, \cdot \rangle$ denotes the dot product. A large $s_j^{(t)}$ means that the j's user can accurately predict questions from t's task. Thus, the dot product enforces that correspondence between a user skills and the required skills for a given task leads to more likely-correct predictions. To incorporate the above in our model, we propose that conditional probability of correctly predicting the outcome of the question $\boldsymbol{Q}_i^{(t)}$ by the forecaster $\mathcal{F}_j$ as follows:

$$P(R_{ij}^{(t)} = A_i^{(t)} | s_j^{(t)}) := P_{ij}^{(t)} = \frac{1}{1 + (|\mathcal{A}_i^{(t)}| - 1)\exp(-\gamma s_j^{(t)})}, \quad (1)$$

where $|\cdot|$ denotes the cardinality of a set and γ is a tuning parameter. Note that when $s_j^{(t)}$ is equal to zero (i.e. unsuitable forecaster for a particular task), the above probability is equal to $\frac{1}{|\mathcal{A}_i^{(t)}|}$. This means that the forecaster cannot do better than chance or is a spammer. On the other hand, when $s_j^{(t)} \to \infty$, the above probability is equal to one and indicates a perfect forecaster; a participant with negative $s_j^{(t)}$ can be considered as an adversarial user. Our goal is to estimate the skills of all forecasters through analyzing the historical performance of the forecasters on past questions whose outcome is known. Upon learning these parameters, future questions of tasks can be referred to suitable forecasters, i.e. those who have large $s_j^{(t)}$ on that task. Moreover, after learning the user skills, we can use the current framework to solve the cold start problem for new tasks [6].

In our formalism, the participants are assumed to be independent. As a result, we might need a considerable amount of performance history to estimate a participant's skill parameters, but geopolitical crowdsourced forecasting generally suffers from data sparsity. In order to tackle this issue, we employ multitask learning in our framework. We assume that the parameters $\boldsymbol{c}_j$ are related and can be represented sparsely in a shared dictionary domain, i.e. $\boldsymbol{c}_j = \boldsymbol{L}\alpha_j$, where $\boldsymbol{L} \in \mathbb{R}^{d\times k}$ and $\alpha_j \in \mathbb{R}^k$ is a sparse vector to ensure that dictionary columns capture a maximal amount of knowledge. This dictionary couples the parameters $\boldsymbol{c}_j$ and allows for using the collective data from all users for more accurate estimates.

3.1 Parameter Inference

We need to infer the parameters $\boldsymbol{s}_j^{(t)}$, $\boldsymbol{L}$, and α_j in our framework. Inspired by the work [9], we use Markov Chain Monte Carlo (MCMC) to infer the parameters $s_j^{(t)}$ using the set $\mathcal{R}$. Having estimated these parameters, we formulate a biconvex optimization problem to infer $\boldsymbol{L}$ and α_j, and hence $\boldsymbol{c}_j$, by using estimated values of $s_j^{(t)}$.

To estimate the values of $s_j^{(t)}$, we use the known $A_j^{(t)}$ and form the posterior distribution for $s_j^{(t)}$ given $R_{ij}^{(t)}$ and $A_i^{(t)}$:

$$P(s_j^{(t)}|R_{ij}^{(t)}, A_i^{(t)}) \propto \prod_{i=1,\, i\in\mathcal{R}}^{n_t} P(R_{ij}^{(t)}|A_i^{(t)}, s_j^{(t)}). \tag{2}$$

Now given the collected data $R_{ij}^{(t)}$ for a fixed j and t and the known outcomes $A_j^{(t)}$, we can compute the maximum a posteriori (MAP) estimate of $s_j^{(t)}$ using Metropolis sampling. To do so, we randomly initialize $s_j^{(t)}$ and then draw a new sample near the previous sample, i.e. $\hat{s}_j^{(t)} \sim \mathcal{N}(s_j^{(t)}, \sigma_M)$, where σ_M is the variance of the jumping distribution. We then compute the posterior probability using Eq. (2) (up to a constant factor). If $P(\hat{s}_j^{(t)}|R_{ij}^{(t)}, A_i^{(t)}) \geq P(s_j^{(t)}|R_{ij}^{(t)}, A_i^{(t)})$, then we update the value of $s_j^{(t)}$, i.e. $s_j^{(t)} = \hat{s}_j^{(t)}$; otherwise, we update the estimate with a given probability P_M. The sampling continues until the estimated value stabilizes, i.e. its estimated value does not change considerably with more sampling.

After estimating the parameters $\{s_j^{(t)}\}$ for all pairs (j, t), and given that $s_j^{(t)} \sim \mathcal{N}(\langle \boldsymbol{d}^{(t)}, \boldsymbol{L}\alpha_j\rangle, \sigma_s)$, we formulate the following objective function to compute MAP estimate of $\boldsymbol{L}$ and α_j as its minimizers:

$$\mathcal{J}(\alpha_j, \boldsymbol{L}) = \sum_{t=1}^{T}\sum_{j=1}^{F} \|s_j^{(t)} - \langle \boldsymbol{d}^{(t)}, \boldsymbol{L}\alpha_j\rangle\|_2^2 + \lambda\|\boldsymbol{L}\|_{\mathsf{F}}^2 + \eta\|\alpha_j\|_1, \tag{3}$$

where $\|\cdot\|_{\mathsf{F}}^2$ denotes the Frobenius norm to regularize complexity of $\boldsymbol{L}$, $\|\cdot\|_1$ denotes the ℓ_1 norm to enforce sparsity on α_j, and λ, η are tunable regularization parameters. The objective function in Eq. (3) is a biconvex function, i.e. it is convex in each variable when the other variable is fixed. In order to solve for its minimizers, we can use alternation on variables and solve the resulting convex problems, allowing us to update the variables iteratively until some convergence criterion is met. When $\boldsymbol{L}$ is fixed, the optimization for α_j reduces to a LASSO problem [5, 15] for each α_j:

$$\hat{\alpha}_j = \arg\min_{\alpha_j} \sum_{j=1}^{F} \|s_j^{(t)} - \boldsymbol{d}^{(t)\top}\boldsymbol{L}\alpha_j\|_2^2 + \eta\|\alpha_j\|_1, \tag{4}$$

where $^\top$ denotes transpose of a matrix.

When all $s_j^{(t)}$'s are fixed, the problem for $\boldsymbol{L}$ has a closed form solution, which can be obtained by nulling the gradient of Eq (3) as:

$$\frac{\partial\mathcal{J}}{\partial\boldsymbol{L}} = \sum_{t=1}^{T}\sum_{j=1}^{F} 2(s_j^{(t)} - \langle \boldsymbol{d}^{(t)}, \boldsymbol{L}\alpha_j\rangle)\boldsymbol{d}^{(t)}\alpha_j^\top + 2\lambda\boldsymbol{L} = 0. \tag{5}$$

To solve for $\boldsymbol{L}$, we vectorize both sides of Eq. (5) and then after applying a property of Kronecker product ($(\boldsymbol{B}^\top \otimes \boldsymbol{A})\mathrm{vec}(\boldsymbol{X}) = \mathrm{vec}(\boldsymbol{AXB})$), Eq. (5) simplifies to the following closed form solution for $\boldsymbol{L}$:

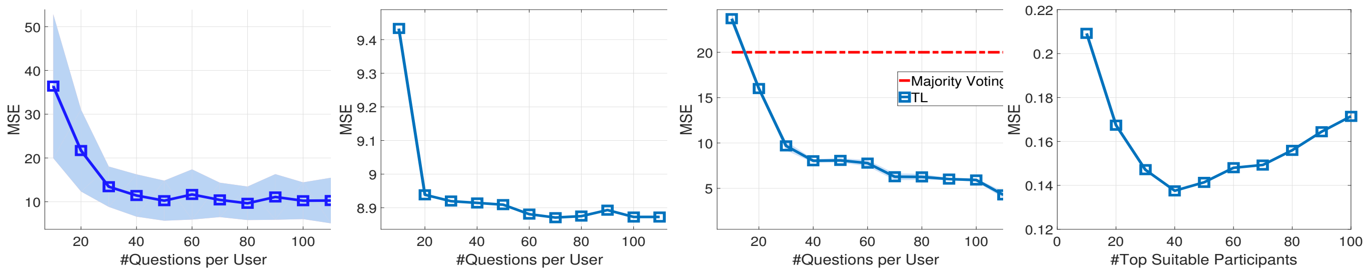
$$\begin{aligned}
\boldsymbol{A} &= \sum_{t=1}^{T}\sum_{j=1}^{F}\left(\alpha_j\boldsymbol{d}^{(t)\top}\alpha_j^\top \otimes \boldsymbol{d}^{(t)\top} + \lambda\boldsymbol{I}_{dk}\right)\\
\boldsymbol{b} &= \sum_{t=1}^{T}\sum_{j=1}^{F}\mathrm{vec}\left(s_j^{(t)}\boldsymbol{d}^{(t)}\alpha_j^\top\right)\\
\boldsymbol{L} &= \mathrm{mat}_{d,k}\left(\boldsymbol{A}^{-1}\boldsymbol{b}\right),
\end{aligned} \tag{6}$$

where vec(·) denotes the matrix to vector and mat(·) denotes the vector to matrix operations. To solve for minimizers of the objective (3), we alternate between Eq. (4) and Eq. (5) until a convergence criterion is met. Upon learning $\boldsymbol{L}$ and α_j, we can compute $\boldsymbol{c}_j$'s.

Having learned all the model parameters, we can recommend a new forecasting question to those who are skilled to answer it, i.e. those with maximum $\langle \boldsymbol{d}^{(t)}, \boldsymbol{c}_j\rangle$ in the data collection step. Moreover, we can incorporate skills of forecasters into the aggregation step by considering the value for probabilities in Eq. (1). As an example, we can use the following aggregation rule:

$$\hat{p}_i^{(t)} = \phi^{-1}\left(\frac{1}{|\mathcal{R}_i^{(t)}|}\sum_{j,\, j\in\mathcal{R}}^{F} \phi(P_{ij}^{(t)})\right), \tag{7}$$

where $\mathcal{R}_i^{(t)}$ denotes the set of the users who have responded to question $Q_i^{(t)}$, and the function $\phi(.\cdot)$ is commonly chosen to be log-odds or the inverse Gaussian function. It has been demonstrated that this aggregation rule results in a more accurate prediction compared to majority voting or a simple average [12].



(a) User Ability Recovery **(b) User Skill Recovery** **(c) MTL vs Majority Voting** **(d) Active vs Passive**

Figure 1: Performance of the proposed algorithm on synthetic dataset. The shaded region shows standard error.

4 EXPERIMENTAL RESULT

In this work, we evaluated the effectiveness of the proposed scheme on a synthetic dataset. The advantage of a synthetic dataset is that we can execute controlled experiments to verify that the algorithm behaves as expected. For future experiments, we are collecting real data in our crowdsourcing data collection online platform.

Our experiment employed five tasks and 100 participants. The number of questions answered by each forecasters in the crowd was varied to analyze the effectiveness of the triage algorithm. We assumed that the questions are binary, $A_i^{(t)} \in \{0, 1\}$, and the true answer for each question is drawn from a symmetric Bernoulli distribution ($Ber(0.5)$). The task parameters $\boldsymbol{d}^{(t)} \in \mathbb{R}^5$ are also assumed to be binary vectors with elements drawn from the Bernoulli distribution. Finally, we also assume that the user skill parameters $\boldsymbol{c}_j \in \mathbb{R}^5$ are drawn from a Gaussian distribution, $\mathcal{N}(3\mathbf{1}_{5\times 1}, \boldsymbol{I}_{5\times 5})$. Given these parameters, we computed $P_{ij}^{(t)}$ for a given user and then randomly set the user prediction synthetically for each question given this probability and the randomly generated true answer $A_i^{(t)}$ of that question, $R_{ij}^{(t)}|A_i^{(t)} \sim Ber(P_{ij}^{(t)})$, to generate the set $\mathcal{R}$. We used the data in each experiment to estimate the model parameters $s_j^{(t)}, \boldsymbol{c}_j$ and then tested the algorithm on the a second set of questions for the same tasks given estimated parameters. We reported the average normalized MSE error (in percentage) on all users in our experiment.

Our results are presented in Figure 1. Figure 1a represents the average quality of recovering the user abilities $s_j^{(t)}$ for all users using MCMC versus the number of questions they answer (for simplicity, we assumed that all users answer an equal number of questions). As expected, as the users answer more questions, their performance history can be used to recover their ability parameters more accurately using MCMC. This experiment indicates that in order to recover ability of a forecaster, we need to collect adequate number of answers for that forecaster. In Figure 1b, we used the estimated parameters $\hat{s}_j^{(t)}$ to solve for the parameters $\boldsymbol{c}_j$ using biconvex optimization. We report the average MSE error of recovering the parameters $\boldsymbol{c}_j$ for the users versus the number of questions each user answers. As expected, similarly the more forecaster participate and provide predictions, the more accurate we can estimate their skills parameters. However, it seems that this effect is weaker compared to recovering the parameters $\hat{s}_j^{(t)}$. This difference can be explained as the result of transfer learning in estimating $\boldsymbol{c}_j$'s. As we share knowledge to recover $\boldsymbol{c}_j$ through estimating the shared dictionary $\boldsymbol{L}$.

In Figure 1c, we illustrate the effect of the proposed scheme on prediction accuracy after aggregation. We compare result of our algorithm which exploits knowledge transfer against majority voting. We compare the prediction accuracy for all test set questions versus the number of training set questions which are used for model parameters recovery. As it can be seen knowledge transfer outperforms naive majority voting considerably. This result suggests that given accurate model parameter estimation our algorithm can be effective for geopolitical forecasting, where the hiring budget is limited and each question can be presented to only a few forecasters or when each forecaster might respond to only a subset of provided questions and hence the response set might be very sparse. Finally, Figure 1d presents results of incorporating the algorithm in a recommender system. We generated a sixth task and referred the questions of the sixth task only to the most top suitable participants (participants with maximum $\boldsymbol{s}_j^{(6)} = \boldsymbol{d}^{(6)\top}\boldsymbol{L}*\alpha_j$ and used predictions of those participants for aggregations. The x-axis denotes the top suitable participants used for aggregation. This figure indicates an important conclusion. As it can be seen, initially the prediction error decreases as we use more suitable participants for aggregation. However, beyond a point, the prediction error increases as more participants are incorporated. This results accords with intuition. In a crowd of participants, a subset are more suitable to predict questions of a task and by identifying this subset, we can improve prediction accuracy (roughly 40 participants in our experiment). The rest of the participants are not skilled enough for the sixth task and hence incorporating their predictions has an outlier effect on aggregation. This result suggests that a suitable recommender system not only can save hiring budget and time of the participants, but can increase the prediction accuracy, i.e. learning more using less amount of data.

5 CONCLUSIONS AND FUTURE WORK

We proposed an MTL algorithm to infer skills of participants in a geopolitical crowdsourced forecasting platform for the purpose of triaging questions to the participants who are most likely to answer them correctly. Experiments on synthetic data demonstrated the effectiveness of our algorithm and paves the way for future experiments on real data. Our ultimate goal is to create an efficient triage framework that allows us to conserve the time of the participants by routing them to the questions that they are most likely to answer correctly while steering away participants who probably will not. The motivation for this is time conservation; as the number of potential forecasting questions increases beyond the capacity for people to address them all, a method like the one we describe here will be required in order to maximize the utility of individual forecasters.

6 ACKNOWLEDGEMENT

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract number 2017-17061500006. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein

REFERENCES

[1] P. D. Atanasov, P. Rescober, E. Stone, S. Swift, E. Servan-Schreiber, P. Tetlock, L. Ungar, and B. Mellers. 2015. Distilling the wisdom of crowds: Prediction markets versus prediction polls. (2015).

[2] Y. Bachrach, T. Graepel, T. Minka, and J. Guiver. 2012. How to grade a test without knowing the answers—a Bayesian graphical model for adaptive crowdsourcing and aptitude testing. *ICML* (2012).

[3] R. Caruana. 1998. Multitask learning. In *Learning to learn*. Springer, 95–133.

[4] M. Dhami, D. Mandel, B. Mellers, and P. Tetlock. 2015. Improving intelligence analysis with decision science. *Perspectives on Psychological Science* 10, 6 (2015), 753–757.

[5] D. Donoho and Y. Tsaig. 2008. Fast solution of ell_1-norm minimization problems when the solution may be sparse. *IEEE Transactions on Information Theory* 54, 11 (2008), 4789–4812.

[6] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong. 2008. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*. ACM, 208–211.

[7] B. Mellers, E. Stone, P. Atanasov, N. Rohrbaugh, S. E. Metz, L. Ungar, M. M. Bishop, M. Horowitz, E. Merkle, and P. Tetlock. 2015. The psychology of intelligence analysis: Drivers of prediction accuracy in world politics. *Journal of experimental psychology: applied* 21, 1 (2015), 1.

[8] B. Mellers, L. Ungar, J. Baron, J. Ramos, B. Gurcay, K. Fincher, S. E. Scott, D. Moore, P. Atanasov, S. A Swift, et al. 2014. Psychological strategies for winning a geopolitical forecasting tournament. *Psychological science* 25, 5 (2014), 1106–1115.

[9] K. Mo, E. Zhong, and Q. Yang. 2013. Cross-task crowdsourcing. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 677–685.

[10] S. Pan and Q. Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2010), 1345–1359.

[11] P. Ruvolo and E. Eaton. 2013. ELLA: An efficient lifelong learning algorithm. In *International Conference on Machine Learning*. 507–515.

[12] V. Satopää, R. Pemantle, and L. Ungar. 2016. Modeling probability forecasts via information diversity. *J. Amer. Statist. Assoc.* 111, 516 (2016), 1623–1633.

[13] A. Spiegel. 2014. So You Think YouâĂŹre Smarter Than A CIA Agent. *NPR parallels blog, April* 2 (2014).

[14] P. Tetlock, B. Mellers, and J. Scoblic. 2017. Bringing probability judgments into policy debates via forecasting tournaments. *Science* 355, 6324 (2017), 481–483.

[15] R. Tibshirani. 2013. The lasso problem and uniqueness. *Electronic Journal of Statistics* 7 (2013), 1456–1490.

[16] J. Whitehill, T. Wu, J. Bergsma, J. Movellan, and P. Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*. 2035–2043.