Reinforcement Peer Prediction: A Reinforcement Learning Framework for Eliciting Information

Zehong Hu
Nanyang Technological University
huze0004@e.ntu.edu.sg

Yang Liu
Harvard University
yangl@seas.harvard.edu

Yitao Liang
University of California, Los Angeles
yliang@cs.ucla.edu

Jie Zhang
Nanyang Technological University
zhangj@ntu.edu.sg

Abstract

Traditional peer prediction mechanisms are designed by assuming an idealized worker model. However, in practice, the worker model is often violated, causing the mechanisms to fail. In this paper, we propose the first model-free reinforcement peer prediction mechanism where a data requester wants to dynamically adjust the incentive level to maximize his revenue. To accurately estimate data requester's revenue and workers' state, we derive their posterior distributions and sample the distributions with Gibbs sampling. Considering the inferred revenue and state are not accurate, we employ online Gaussian process regression to learn the revenue corresponding to different incentive levels and utilize the classic ϵ-greedy strategy to choose the optimal incentive level. The experiments on different worker models show that our mechanism can significantly increase data requester's revenue.

1 Introduction

Crowdsourcing, which elicits useful information from a crowd of online human workers, has arisen as a promising method to facilitate the development of machine learning systems. For example, a popular application of crowdsourcing is to generate labels for large scale machine learning datasets such as RTE [14] and ImageNet [5]. Notwithstanding its high efficiency and immediate availability, one salient concern about crowdsourcing is the quality of collected answers, because it is often difficult or too costly to verify workers' answers. This problem is called information elicitation without verification [16]. A class of incentive mechanisms, collectively called peer prediction, has been developed to solve this problem [11, 8, 19, 18, 12]. In peer prediction, the payment for one worker is decided by comparing his answers with those of his peers, and the payment rules are elegantly designed so that truthful report is a game-theoretic equilibrium for all workers.

Effort elicitation is another goal considered by more recent peer prediction mechanisms [17, 3, 13, 9]. In these mechanisms, workers are incentivized not only to report truthfully, but also to generate high quality answers by exerting their maximal efforts. For all the aforementioned peer prediction mechanisms, one essential precondition is an explicitly-known worker model which includes workers' utility function and the assumption that workers are fully rational and only take the utility-maximizing strategy. However, in practice, workers may be bounded rational [10, 7]. Or, they do not calculate the utility-maximizing strategy but gradually adjust their strategy according to the interaction with the peer prediction mechanism [2, 15]. In some cases, workers' utility function may also be different from our anticipation [1]. To avoid this precondition, we propose to decide the payment for workers by using reinforcement learning. It learns the optimal payment based on workers' contributions at each step. Nevertheless, there are two main challenges. Firstly, classic reinforcement learning focuses

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

on the interaction between a single agent and its environment. It does not consider the game among workers, and thus may violate the incentive compatibility requirement that rational workers should be able to gain the maximal utility by reporting truthfully and exerting maximal efforts. Secondly, no ground-truth answers are available for evaluating the benefits of workers. Hence, we need to find a proper way to compute workers' contributions so that reinforcement learning can be applied.

The main contributions of this paper are the following four aspects. Firstly, we propose the first model-free reinforcement peer prediction mechanism. Our mechanism uses reinforcement learning to calculate the overall incentive level, traditional peer prediction mechanisms to decide the payment for each worker, and Bayesian inference to evaluate workers' contributions. Secondly, classic Bayesian inference algorithms for crowdsourcing (e.g. the Dawid-Skene estimator [4]) suffer from local optimum, causing the estimation of workers' contributions to be biased. This bias will further mislead reinforcement learning. Thus, we derive the explicit posterior distribution of workers' contributions and employ Gibbs sampling for inference to eliminate the bias. Thirdly, in our mechanism, the inferred contributions are corrupted by noise and we can only observe the last step worker state rather than the current one. Hence, in reinforcement learning, we use the online Gaussian process regression to learn the Q-function and replace the unknown current state with the couple of the last step state and incentive level. Fourthly, we conduct empirical evaluation on different worker models, which shows that our mechanism can significantly increase the revenue of using crowdsourcing.

2 Problem Formulation

Suppose in our system there are one data requester and N candidate workers denoted by $\mathcal{C} = \{1, \ldots, M\}$, where $M \geq 4$. At every stage, the data requester assigns M binary answer tasks, with answer space $\{1, 2\}$, to workers. At stage t, for task i, worker j's label can be written as $L_t(i, j)$, and correspondingly our mechanism needs to pay $P_t(i, j)$. Besides, we use $L_t(i, j) = 0$ to denote that task i is not assigned to worker j, and thus $P_t(i, j) = 0$. After collecting labels, the data requester will aggregate labels via Bayesian inference [20], and the label accuracy can be written as A_t. Thus, the revenue of the data requester at stage t can be calculated as:

$$r_t = F(A_t) - \eta \sum_{i=1}^{N} \sum_{j=1}^{M} P_{ij} \tag{1}$$

where $F(\cdot)$ maps accuracy into revenue. In practice, the higher accuracy, the better. Meanwhile, the collected labels can only be used when their accuracy reaches a certain requirement. Thus, we set $F(A_t) = A_t^n$ in this paper. It is worth noting that our reinforcement peer prediction mechanism does not require any specific formulation of the F function. Suppose our mechanism goes for T stages. Thus, the goal of our mechanism is to maximize the accumulative revenue, namely $R = \sum_{t=1}^{T} r_t$.

3 Reinforcement Peer Prediction

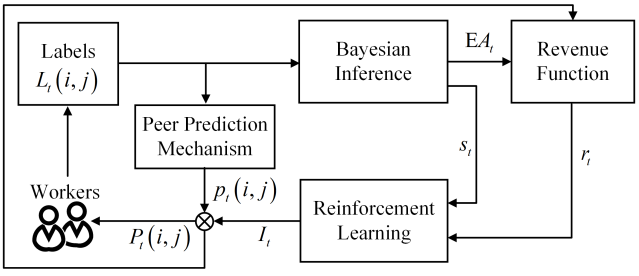


Figure 1: The architecture of our reinforcement peer prediction mechanism

We present the architecture of our mechanism in Figure 1. At step t, our mechanism decides the payment for workers as $P_t(i, j) = I_t \cdot p_t(i, j)$, where I_t denotes the incentive level calculated by our reinforcement learning algorithm. $p_t(i, j)$ denotes the output of the peer prediction mechanism. By doing so, we can ensure the reinforcement learning does not break the incentive compatibility requirement, that is, truthful report and exerting maximal efforts is always a Nash equilibrium for all workers. Besides, we use Bayesian inference to aggregate the collected labels. Since the ground-truth

answers are unavailable, we cannot directly calculate the accuracy A_t. Thus, we use the expected accuracy $\mathbb{E}A_t$ instead. It can be calculated as $\mathbb{E}A_t = \frac{1}{N}\sum_{i=1}^{N}\Pr(L_i = y_i)$, where L_i and y_i denote the aggregated and true label, respectively. Meanwhile, Bayesian inference can also output the confusion matrices of all workers and the distribution of task labels $[\Pr(l=1), \Pr(l=2)]$. For worker j, his confusion matrix $C_j = [c_{jkg}]_{2\times 2}$, where c_{jkg} denotes the probability that worker j labels a task in class k as class g. Since the accuracy of labels are determined by the overall quality of workers, how an individual worker performs is not the concern of our reinforcement learning algorithm. Thus, we denote the state of the whole worker crowd s_t by workers' average probability of being correct, namely $s_t = \sum_{k\in\{1,2\}}\Pr(l=k)\cdot\frac{1}{M}\sum_{j=1}^{M}c_{jkk}$. After workers receive the payment, they may adjust their strategies, which will lead to the change s_t. However, when computing the incentive level I_{t+1}, we only has the last step state s_t. In other words, the state observation has one-stage delay, which makes our reinforcement learning problem different from traditional ones.

Peer Prediction Mechanism: We adopt the state-of-the-art mechanism proposed by [3]. For each task-worker pair (i,j), it selects a reference worker k. Suppose workers j and k have been assigned d other distinct tasks $\{j_1,\ldots,j_d\}$ and $\{k_1,\ldots,k_d\}$, respectively. Then, the payment $p_t(i,j) = 1[L(i,j)=L(i,k)] - \xi_j^d\cdot\xi_k^d - \bar{\xi}_j^d\cdot\bar{\xi}_k^d$, where $\xi_k^d = \sum_{g=1}^{d}1(L(i_g,k)=1)/d$ and $\bar{\xi}_k^d = 1-\xi_k^d$.

Bayesian Inference: Suppose the prior distributions that $c_{jk1}\sim\text{Beta}(\alpha_{jk1}^0,\alpha_{jk2}^0)$ and $\Pr(l=1)\sim\text{Beta}(\beta_1^0,\beta_2^0)$. Then, we can explicitly derive the posterior distribution of true labels as

$$P(\boldsymbol{y}|\boldsymbol{L}) \propto B(\boldsymbol{\beta})\prod_{j=1}^{M}\prod_{k=1}^{K}B(\boldsymbol{\alpha}_{jk})\ ,\ \alpha_{jkg} = \sum_{i=1}^{N}\delta_{ijg}\xi_{ik} + \alpha_{jkg}^0, \beta_k = \sum_{i=1}^{N}\xi_{ik} + \beta_k^0 \quad (2)$$

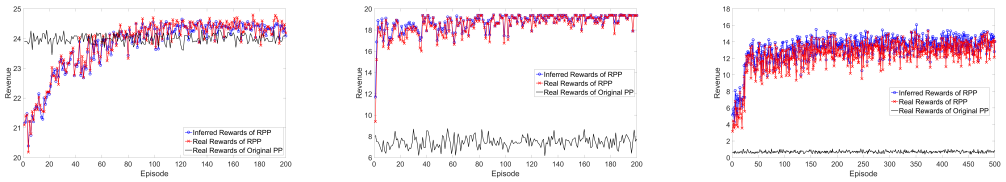
where $B(\cdot)$ denotes the beta function, $\delta_{ijg} = 1(L(i,j)=g)$ and $\xi_{ik} = 1(y_i = k)$. According to Gibbs sampling, we can generate posterior samples via iteratively sampling $P(y_i|\boldsymbol{L},\boldsymbol{y}_{j\neq i})$.

Reinforcement Learning: When calculating the incentive level I_t for stage t, the current state s_t cannot be observed. Thus, we define our incentive policy as $\pi(I_t|x_t)$, where $x_t = \langle s_{t-1}, I_{t-1}, t\rangle$. Then, the Q-function of our policy can be calculated as $Q(x_t,I_t) = \sum_{i=0}^{T-t}\gamma^i r_{t+i}$, where the discount factor γ is slightly smaller than 1 and $Q(x_{T+1},I_{T+1}) = 0$. Both the state s_t and reward r_t are not accurately observed. Thus, we apply Gaussian process regression for the Q-function:

$$Q(x_t,I_t) - \gamma Q(x_{t+1},I_{t+1}) = r_t + N(x_t,x_{t+1}) \quad (3)$$

where the residual $N(x_t,x_{t+1})$ is assumed to be a Gaussian process. By applying the online Gaussian process regression algorithm [6], we can learn the Q-function. Then, we decide the incentive level I_t for stage t as $\arg\max Q(x_t,I_t)$ with probability $1-\epsilon$ and a random value with probability ϵ.

4 Experiments



(a) Rational Workers (b) Quantal Response Workers (c) MWUA Workers

Figure 2: Experiments on different worker models

In Figure 2, we present the simulation results on different worker models. In our experiments, we assume there are four incentive levels, namely $I_t \in \{0.1, 1.0, 5.0, 10.0\}$. Meanwhile, we set the number of workers and tasks at each stage as 10 and 100, respectively. For rational workers, we assume all workers report the true labels with probability 0.9 for all incentive levels. For quantal response workers, we assume workers use the quantal response model [10] to decide whether to exert the maximal efforts. For MWUA workers, we assume workers follow the MWUA model [2]. From all the experiments, we can find that our reinforcement peer prediction mechanism can gradually lean the optimal incentive strategy, which significantly increase the revenue when workers are not rational.

References

[1] D. Bergemann, S. Morris, et al. An introduction to robust mechanism design. *Foundations and Trends® in Microeconomics*, 8(3):169–230, 2013.

[2] E. Chastain, A. Livnat, C. Papadimitriou, and U. Vazirani. Algorithms, games, and evolution. *PNAS*, 111(29):10620–10623, 2014.

[3] A. Dasgupta and A. Ghosh. Crowdsourced judgement elicitation with endogenous proficiency. In *Proc. of WWW*, 2013.

[4] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. of CVPR*, 2009.

[6] Y. Engel, S. Mannor, and R. Meir. Reinforcement learning with gaussian processes. In *Proc. of ICML*, 2005.

[7] R. Jurca and B. Faltings. Robust incentive-compatible feedback payments. In *Agent-Mediated Electronic Commerce. Automated Negotiation and Strategy Design for Electronic Markets*, pages 204–218. Springer, 2007.

[8] R. Jurca, B. Faltings, et al. Mechanisms for making crowds truthful. *Journal of Artificial Intelligence Research*, 34(1):209, 2009.

[9] Y. Liu and Y. Chen. Sequential peer prediction: Learning to elicit effort using posted prices. In *AAAI*, pages 607–613, 2017.

[10] R. D. McKelvey and T. R. Palfrey. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38, 1995.

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

[12] G. Radanovic and B. Faltings. A robust bayesian truth serum for non-binary signals. In *Proc. of AAAI*, 2013.

[13] V. Shnayder, A. Agarwal, R. Frongillo, and D. C. Parkes. Informed truthfulness in multi-task peer prediction. In *Proc. of ACM EC*, 2016.

[14] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proc. of EMNLP*, 2008.

[15] H. Stojic, P. P. Analytis, and M. Speekenbrink. Human behavior in contextual multi-armed bandit problems. In *Proc. of CogSci*, 2015.

[16] B. Waggoner and Y. Chen. Output agreement mechanisms and common knowledge. In *Proc. of HCOMP*, 2014.

[17] J. Witkowski, Y. Bachrach, P. Key, and D. C. Parkes. Dwelling on the negative: Incentivizing effort in peer prediction. In *Proc. of HCOMP*, 2013.

[18] J. Witkowski and D. C. Parkes. Peer prediction without a common prior. In *Proc. of ACM EC*, 2012.

[19] J. Witkowski and D. C. Parkes. A robust bayesian truth serum for small populations. In *Proc. of AAAI*, 2012.

[20] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng. Truth inference in crowdsourcing: is the problem solved? *Proc. of the VLDB Endowment*, 10(5):541–552, 2017.