Complex crowdsourcing task allocation strategies employing supervised and reinforcement learning

Received 10 August 2017
Revised 4 September 2017
Accepted 5 September 2017

Complex crowdsourcing task allocation strategies employing supervised and reinforcement learning

Lizhen Cui
School of Computer Science and Technology, Shandong University, Jinan, China, and School of Software Engineering, Shandong University, Jinan, China

Xudong Zhao
School of Software Engineering, Shandong University, Jinan, China

Lei Liu
School of Computer Science and Technology, Shandong University, Jinan, China

Han Yu
Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY) Nanyang Technological University Singapore, Singapore, and

Yuan Miao
Victoria University, Melbourne, Australia

Abstract

Purpose – Allocation of complex crowdsourcing tasks, which typically include heterogeneous attributes such as value, difficulty, skill required, effort required and deadline, is still a challenging open problem. In recent years, agent-based crowdsourcing approaches focusing on recommendations or incentives have emerged to dynamically match workers with diverse characteristics to tasks to achieve high collective productivity. However, existing approaches are mostly designed based on expert knowledge grounded in well-established theoretical frameworks. They often fail to leverage on user-generated data to capture the complex interaction of crowdsourcing participants' behaviours. This paper aims to address this challenge.

Design/methodology/approach – The paper proposes a policy network plus reputation network (PNRN) approach which combines supervised learning and reinforcement learning to imitate human task allocation strategies which beat artificial intelligence strategies in this large-scale empirical study. The proposed approach incorporates a policy network for the selection of task allocation strategies and a

This work is partially supported by NSFC (No.61572295), the Innovation Method Fund of China (No.2015IM010200), SDNSFC (No. ZR2014FM031), the TaiShan Industrial Experts Programme of Shandong Province, the Shandong Province Science and Technology Major Special Project (No.2015ZDXX0201B03, 2015ZDXX0201A04, 2015ZDJQ01002), the Shandong Province Key Research and Development Plan (No.2015GGX101015, 2015GGX101007).

reputation network for calculating the trends of worker reputation fluctuations. Then, by iteratively applying the policy network and reputation network, a multi-round allocation strategy is proposed.

Findings – PNRN has been trained and evaluated using a large-scale real human task allocation strategy data set derived from the Agile Manager game with close to 500,000 decision records from 1,144 players in over 9,000 game sessions. Extensive experiments demonstrate the validity and efficiency of computational complex crowdsourcing task allocation strategy learned from human participants.

Originality/value – The paper can give a better task allocation strategy in the crowdsourcing systems.

Keywords Crowd behaviour analysis, Task-oriented crowdsourcing

Paper type Research paper

1. Introduction

Recently, crowdsourcing system is an emerging platform for completing tasks from crowds. It has been increasingly popular especially for the tasks that are quite difficult and challenging for computers. They are popular especially for the tasks that are too challenging for complete automation such as sentiment analysis (Aljanaki *et al.*, 2016; Liu *et al.*, 2012), biomedical research (Khare *et al.*, 2015) and software engineering (LaToza and van der Hoek, 2016).

Requesters and *workers* are the two critical roles in a typical crowdsourcing system. Requesters post tasks to the system, whereas workers choose which task requests to accept and complete the selected tasks. It is common for tasks with high difficulty or involve heavy workload to be decomposed into smaller and simpler human intelligent tasks (HITs) so that each HIT can be completed by a worker over a relatively shorter period of time. Workers' abilities may vary significantly, resulting in different quality of results for the completed tasks. Different tasks may also carry different rewards according to factors such as their difficulties and workloads.

Under the current mode of operation, requesters first post their tasks to a crowdsourcing system. Workers then browse the available tasks and select the ones they are interested in. Finally, the requesters determine whether to give the workers rewards based on the quality of the completed tasks. Existing algorithmic crowdsourcing approaches focus on designing mechanisms to match tasks to suitable workers taking into account factors such as their skill sets, reputation, availability and current workload. Such approaches are either based on recommendations (i.e. telling wor kers what to do) (Ho and Vaughan, 2012; Nath and Narayanaswamy, 2014), (Yu *et al.*, 2013, 2016, 2015a, 2015b, 2015c, 2013) or incentives (i.e. influencing workers' behaviours through economic means) (Liu *et al.*, 2015; Miao *et al.*, 2016; Yu *et al.*, 2015a, 2015b, 2015c), (Tran-Thanh *et al.*, 2014, 2015). However, these approaches depend heavily on expert knowledge in the sense that the mechanisms are designed by experts through analysing the specific scenarios facing crowdsourcing under well-established theoretical frameworks such as queueing theories. As human behaviours in large-scale interactions stimulated by monetary incentives can be highly sophisticated, they cannot be completely captured by these theoretical frameworks.

To address this problem, we propose a machine learning based approach to enable agents to learn from human task allocation strategies with good performance. The learning approach combines supervised learning with reinforcement learning to obtain near-optimal solutions. Specifically, we first formulate the complex task allocation problem as an optimisation problem. As it consists of two neural networks: the *policy network* and the *reputation network*, we name the proposed approach the *policy network plus reputation network* (PNRN) approach. The policy network explores for better task allocation strategy based on the total reward return over all successive allocations based on Q-learning, whereas the reputation network is trained by supervised learning with the real human task allocation strategy data to predict the trends of reputation fluctuations. Next, an iterative between policy and reputation network is applied to obtain a multi-round allocation strategy. Training was conducted with an empirical data set containing close to 500,000 decision records from 1,144 players in over 9,000 game sessions using the *Agile Manager* game platform (Yu *et al.*, 2014) which was released in Yu *et al.* (2017).

By learning from human decisions which play against the artificial intelligence (AI) powered crowdsourcing task allocation approach (Yu *et al.*, 2013) from this unique data set, the PNRN has been shown to be able to form efficient multi-round complex task allocation strategies. PNRN has been implemented as a decision support agent and incorporated into the Agile Manager game platform. Through extensive competition with the existing AI approach in the platform, the strategy learnt by PNRN from human players has been shown to match the performance of (Yu *et al.*, 2013) which has previously beat human players more than $\frac{2}{3}$ of the time.

PNRN advances the state-of-the-art in algorithmic crowdsourcing by enabling future task allocation approaches to learn from human behaviours. We plan to further extend our current study to other crowdsourcing platforms to investigate how well the PNRN approach can generalize in different crowdsourcing situations.

2. Related work

Task allocation in crowdsourcing system has been an interesting and challenging problem. A number of existing works focusing on reputation-based task allocation have been proposed. In these works, reputation values are often calculated from workers' historical performance data (jagabathula *et al.*, 2014; Jøsang *et al.*, 2007; Man *et al.*, 2011). The basic idea of these works is to provide a reasonable estimation on the reputation values of the workers, based on which task allocation approaches can be designed. (Jagabathula *et al.*, 2014) designed a computationally efficient reputation algorithm to identify and filter out adversarial workers in crowdsourcing system. Man *et al.* (2011) proposed a robust training algorithm using fuzzy neural approaches to evaluate reputation. By improving the accuracy of reputation evaluation, tasks can be allocated to the workers with high reputation in an attempt to obtain high quality results.

With respect to task allocation, in Macarthur *et al.* (2011), the authors proposed a distributed anytime algorithm for task allocation where the tasks and agents constantly change over time. Yu *et al.* (2013) proposed an approach for workers to make situation-aware task acceptance decisions by jointly considering their current reputation and workloads to avoid large fluctuations in their perceived reputation. Kartal *et al.* (2016) proposed a Monte Carlo Tree Search based algorithm to solve the task allocation problem with spatial, temporal and other constraints.

Currently, approaches designed to solve the crowdsourcing task allocation problem rely heavily on expert knowledge. Thus, they are limited to what we have already understood about human behaviour under these conditions and are unable to leverage on the large amount of behavioural trajectory data available in crowdsourcing platforms. The PNRN approach proposed in this paper bridges this gap and serves as a framework to enable future crowdsourcing systems to incorporate decision support agents capable of learning from human decision strategies to improve task allocation decisions iteratively.

3. Problem formulation

In this section, we formulate the crowdsourcing task allocation problem to provide a foundation based on which PNRN can be proposed.

3.1 Modelling workers and tasks

In essence, task allocation in a crowdsourcing system can be modelled as a series of interactions between a decision support agent (a.k.a. the system) and the workers. The system dispatches certain tasks to workers. The workers then complete the assigned tasks at certain quality level. There are two important elements in such a process: *tasks* and *workers*. Both of the two factors can be heterogeneous with different characteristics.

To formulate the task allocation problem, we firstly study the attributes of both tasks and workers, which can be regarded as inputs for the task allocation problem. According to the characteristics of general crowdsourcing system (Yu *et al.*, 2012), the basic attributes of tasks and workers are summarised in Table I and Table II, respectively.

Assume that there are T tasks, $\{t_1, t_2, \ldots, t_T\}$, to be allocated to W workers, $\{w_1, w_2, \ldots, w_W\}$. According to Table I and Table II, tasks and workers have their own attributes which can be denoted as equation (1) and (2):

$$\{V, D, SR, ER, DL\} \tag{1}$$

$$\{R, S, EUT, CT, EC\} \tag{2}$$

3.2 Objective function

In crowdsourcing, allocation of a group of at least one task can be modelled as an allocation chain with a length of at least 1.

The reward that a worker can eventually obtain depends on the quality of the completed tasks. Specially, the worker w_j is awarded with the agreed reward V_i for task t_i if and only if

Attribute	Description
Value (V)	Represents the reward for a task completed with high quality by a worker
Difficulty (D)	Represents the complexity of a task
Skill required (SR)	Specifies the capability required for a worker to perform a task
Effort required (ER)	Specifies the workload imposed by a task
Deadline (DL)	Specifies the point in time before which a task must be completed

Table I. Tasks attributes

Attribute	Description
Reputation (R)	An estimation on the ability of a worker, which can be expressed as the probability that the worker is able to complete a task with high quality
Skill (S)	Capabilities that a worker possesses
Effort per unit time (EUT)	Specifies the maximum productivity of a worker per unit time
Current tasks (CT)	Represents current total tasks allocated to a worker
Effort consumed (EC)	Represents current effort consumed by tasks allocated to a worker It equals to the sum of effort required for each task in a worker's CT backlog

Table II. Workers attributes

the task is completed by w_j with high quality. The reward gained by a worker, g_j^i, can be expressed as:

$$g_j^i = \begin{cases} V_i, \ \textit{if } t_i \textit{ is completed with high quality}; \\ 0, \ \textit{otherwise} \end{cases} \tag{3}$$

Here, "high quality" means the following two necessary conditions must be satisfied.

The first necessary condition is that the task should be completed strictly no later than its deadline. In this paper, we assume that the tasks are serviced by a worker on a first-come-first-served basis, and the maximum effort e_j of the worker w_j is proportional to the period of time, DL_i, before the deadline of the task. Such relationship can be expressed by equation (4). Then, g_j^i can be re-written as equation (5), where g_j^i will be 0 if the sum of the current effort consumed by the pending tasks, EC_{CT}, and the effort required, ER_i, is more than worker w_j's maximum effort e_j:

$$e_j = EUT_j \times DL_i \tag{4}$$

$$g_j^i = 0, \text{ if } EC_{CT} + ER_i > e_j \tag{5}$$

The second necessary condition is that the task must be completed correctly, which is closely related to the difficulty and the ability of the worker. The probability for any given worker to achieve the two aforementioned conditions can be expressed by the reputation value which has been well studied in (Yu *et al.*, 2014).

Our objective is to efficiently allocate a given group of heterogeneous tasks to reliable workers to obtain as much total reward, r_{tot}, as possible. This problem can be formulated by equations (6)-(9).

Maximize:

$$r_{tot} = \sum_{j=1}^{W} \sum_{i=1}^{T} g_j^i \tag{6}$$

subject to:

$$0 \le CT_j \le T, \ \ j \in [1, W] \tag{7}$$

$$EC_j \ge 0, \quad j \in [1, W] \tag{8}$$

$$\sum_{j=1}^{W} EC_j = \sum_{i=1}^{t} ER_i, \ \ t \in [1, T] \tag{9}$$

The objective is to maximize the global reward r_{tot}. Meanwhile, three constraints must be satisfied. Constraints equations (7) and (8) are the limitations on the workers' backlog queue lengths and their productivity, respectively. Constraint equation (9) dictates the effort expended by all workers should be equal to the effort required by all allocated tasks. Note that each task can be allocated to only one worker in our paper.

4. Policy network plus reputation network approach

With the aforementioned problem formulation, we now proceed to propose the PNRN approach.

4.1 Pre-treatment and preparation

According to Table I, heterogeneous tasks may have different requirements, especially skills required. To ensure that the tasks can be allocated to the workers who have the corresponding skills, we assume that skills required by the tasks and the skills that the workers possess are both known to the crowdsourcing system. This assumption is reasonable because the requesters can label their tasks with this attribute, and the workers can indicate their skills to the crowdsourcing system. Therefore, we firstly classify all tasks into different groups, tasks under the same group require the same skill.

With regard to a given skill S_a, tasks and workers are divided into different groups G_{t_Sa} and G_{w_Sa}. Considering that the characteristics of the tasks and the workers may vary over time, the proposed approach dynamically determines the number of tasks and workers in each group during the classification process.

For tasks, the number of tasks is fixed to $n\_ task$, to simplify the problem. We also fixed the proportion and location of different tasks in a group based on the values of tasks. A task with a given value is located in the corresponding area in a group. As shown in Figure 1, tasks with value V_i are located in the area p_{i_1} to p_{i_b}, where b represents the maximum number of this type task in this group. If there are more than $n\_ task$, the remaining tasks are grouped continually until there remains no more than $n\_ task$ tasks. As the number of leftover tasks is no more than $n\_ task$, the corresponding location can be replaced with zero.

For workers, the number of workers is fixed to $n\_ worker$ with preference given to workers with high reputation values (i.e. good track records). If the optional workers are so larger than the number is more than n_{worker}, meanwhile, the workers can be selected randomly or based on the reputation of the worker. We suggest that the workers be selected based on the reputation because the reputation represents the past performance of the worker, and this approach conforms to the selection of human, which can be explained from the real human decision-making in section of Experimental Evaluation.

4.2 Policy network

Algorithm 1 Learning the Policy Network

```
1:  Initialize action-value function $Q$
2:  for $episode = 1$ to $episode_{max}$ do
3:    initialize state $s_1$
4:    initialize workers' reputation $\{R_1, \ldots, R_{n\_worker}\}$;
5:    for $k = 1$ to $n\_task$ do
6:        With probability $\varepsilon$ select a random action $a_k$;
7:        otherwise select $a_k = \max_a Q^*(s, a; \theta)$;
8:        Execute action $a_k$
9:        if $EC_{k-1} + ER_k > e_j$ then
10:         $r_k \leftarrow {g_j}^k \leftarrow 0$;
11:       else
12:         if $R_j \geq random(0,1)$ then
13:           $r_k \leftarrow g_j^k \leftarrow V_k$;
```

```
      else
        $r_k \leftarrow g_j^k \leftarrow 0$;
      end if
    end if
    if Current state is the terminal state $s_{k+1}$ then
      $y_k \leftarrow r_k$;
    else
      $y_k \leftarrow r_k + \lambda \max_{a'} Q(s_{k+1}, a'_{k+1}; \theta)$;
    end if
    Calculate the loss function according to equation (11);
    Perform gradient descent according to equation (12);
  end for
end for
```

In terms of task allocation, the crowdsourcing system can be viewed as an agent. The agent allocates complex tasks in a sequence of actions, observations and rewards. The complex tasks are allocated in sequence, and the states, $\{s_1, s_2, \ldots, s_{n\_task+1}\}$, represent whether each task is allocated. The final state represents the end of task allocation. At step k, the agent selects an action a_k from the set of legal allocation actions, $A = \{1, 2, \ldots, n\_worker\}$, which represents that a task can be allocated to any worker. An action is selected at a state, and the agent proceeds to the next state. The reward is the value of the task. However, whether this reward can be obtained by a worker depends on the result produced by the worker. The state transition of the task allocation process is shown in Figure 2.

To achieve the objective expressed by equation (6), i.e. to select a policy of actions for maximizing the future rewards, we apply the Q-learning (Watkins and Dayan, 1992) to explore for better policies. Assuming that the future rewards are discounted by a factor of λ per time step, we want to obtain the optimal action-value function $Q^*(s, a)$, which satisfies the following equation:

$$Q^*(s,a) = E_{s'}\left[r + \lambda \max_{a'} Q^*(s',a')|s,a\right] \tag{10}$$

We use a neural network function approximator with weights θ, which we name as the policy network, to estimate the action-value function. The neural network consists of one

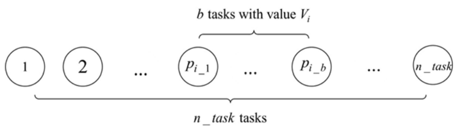

Figure 1.
The location of tasks in a group

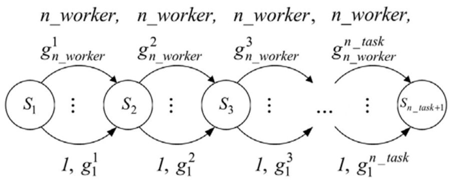
n_worker, $g^1_{n\_worker}$
n_worker, $g^2_{n\_worker}$
n_worker, $g^3_{n\_worker}$
n_worker, $g^{n\_task}_{n\_worker}$
S_1
S_2
S_3
...
S_{n_task+1}
$1, g^1_1$
$1, g^2_1$
$1, g^3_1$
$1, g^{n\_task}_1$

Figure 2.
Task allocation state transition

convolutional layer and one sub sampling layer. The policy network is trained by minimizing the loss function $L(\theta)$:

$$L(\theta)=E\left[\left(r+\lambda\max_{a^t}Q\left(s^j,a^j;\theta\right)-Q(s,a;\theta)\right)^2\right] \tag{11}$$

Therefore, the gradient of the loss function can be calculated as follows:

$$\frac{\partial L(\theta)}{\partial\theta}=E\left[\left(r+\lambda\max_{a^t}Q\left(s^j,a^j;\theta\right)-Q(s,a;\theta)\right)\frac{\partial Q(s,a;\theta)}{\partial\theta}\right] \tag{12}$$

The stochastic gradient decent approach is used to optimize the loss function. Based on the problem formulation and above description of policy network, the algorithm for learning the policy network is shown in Algorithm 1.

The inputs to the policy network include all $n\_$ *worker* workers' current reputation, which can be denoted as equation (13):

$$\{R_1,R_2,R_3,\ldots,R_{n\_worker}\} \tag{13}$$

The outputs to the policy network are the task allocation strategies of all $n\_$ *task* tasks, which can be denoted as equation (14):

$$\{a_1,a_2,\ldots,a_k,\ldots,a_{n\_task}\},a_k\in A \tag{14}$$

4.3 Reputation network

Reputation is computed based on workers' past performance. Thus, it is important to learn the trend of reputation fluctuations based on the current reputation and task allocation strategies. For this purpose, we build a reputation network based on a back-propagation neural network (BPNN) (Goh, 1995) to estimate the trend of reputation variation. The three-layer BPNN has been identified as the best performer (Chen *et al.*, 2012). Therefore, the reputation network also adopts the three-layer structure.

To improve the accuracy of the reputation network, the network should be trained with real human task allocation strategies, which can be obtained from the historical records of crowdsourcing systems.

The inputs to reputation network include the current reputation R of all $n\_$ *worker* workers and the task allocation strategies a for allocating all $n\_$ *task* tasks to the $n\_$ *worker* workers, which can be denoted as equation (15):

$$\{R_1,R_2,\ldots,R_{n\_worker},a_1,a_2,\ldots,a_{n\_task}\} \tag{15}$$

The output of the network is the estimated reputation fluctuation R' of all $n\_$ *worker* workers, which can be denoted as equation (16):

$$\{R'_1,R'_2,\ldots,R'_{n\_worker}\} \tag{16}$$

Because the unit and range of the inputs and the outputs are different, normalisation is necessary before creating and training the reputation network. We adopt the minmax method to normalize a given variable x ($x \in [x_{\min}, x_{\max}]$) to the interval of $[y_{\min}, y_{\max}]$. The normalisation process is carried out as follows:

$$y = (y_{\max} - y_{\min})\frac{x - x_{\min}}{x_{\max} - x_{\min}} + y_{\min} \tag{17}$$

Because the number of neurons in the hidden layer influences the accuracy and effectiveness of the reputation network, the selection of the number of neurons will be studied in the following section. After selecting the number of neurons through training with the real-word data set, the reputation network can be used to compute the changes in workers' reputations.

4.4 Iteration between policy and reputation networks

The task allocation strategy obtained from the policy network only based on the current reputation may be worse. Therefore, a multi-round allocation strategy is proposed which iteratively applying the policy network and reputation network. The iteration between policy network and reputation network can be explained by Figure 3.

As shown in Figure 3, a task allocation strategy can be obtained according to the current reputation of workers through the policy network. Then, by using the reputation network, the subsequent reputation of the workers can be estimated based on their current reputation and the current task allocation strategy. Finally, the above two processes are iterated several times to simulate multi-round allocation and explore a better allocation strategy. The selection of the number of iterations will be studied in the following section.

5. Experimental evaluation

In this section, we evaluate the effectiveness of the proposed approach.

5.1 Real-world data set

We use the Agile Manager data set to train and evaluate the proposed approach. The data set is based on the Agile Manager game (Yu *et al.*, 2014), which is a platform that tests real users' strategies for allocating a given set of tasks to worker agents over multiple rounds against an AI approach from (Yu *et al.*, 2013). The game score is the sum of reward obtained by the worker agents. A total of 30 tasks and 10 worker agents are specified in the game. The game contains six game levels, the tasks and worker agents have heterogeneous characteristics which also vary across different game levels. The game interface is shown in Figure 4. It is able to present the decision choice options to players, capture their actual task allocation decisions and iteratively reveal more information to the players to simulate information uncertainty in practice. For more detailed descriptions of the game platform, please refer to Yu *et al.* (2014, 2015a, 2015b, 2015c, 2017).

The data set contains 9,854 game sessions and 495,533 task allocation decision records by 1,144 human players. After each game session, the player is required to report to the

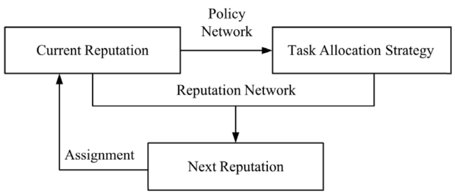

Figure 3.
The iteration between the policy network and the reputation network

strategy used, which can be chosen a mixture of available options, during the game session. Figure 5 shows the distribution of the human task allocation strategies. As can be seen from Figure 5, reputation-based task allocation strategy is the most widely adopted strategy. Meanwhile, the load-balancing is also one of the important factors that human players consider. These two dominant factors correspond to the two conditions that we consider to obtain high quality results in this paper. Overall, the game AI is the best performing strategy, beating player strategies 67.43 per cent of the time.

During pre-treatment, we set the number and sequence of tasks and workers in the group in our solution in accordance to the game. The concrete task allocation strategy can be extracted from the game records. We focus on using the decision data records belong to the 32.57 per cent of the game sessions in which human strategies outperformed the game AI to train PNRN.

5.2 Experimental results

The policy network is a convolutional neural network, trained with the Q-learning, with the current reputation of the 10 worker agents as inputs and the task allocation policy with the highest score as the output. In the process of training, whether the score is obtained or not is based on the reputation and the judgement described in equation (5), which is also shown in Algorithm 1. It should be noted that the network is firstly trained by the human allocation strategies that play against the AI to obtain the initialised action-function Q in the algorithm.

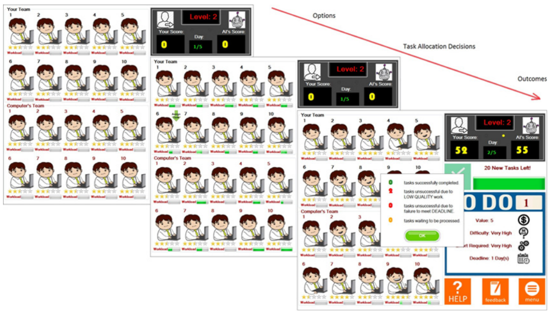

Figure 4. The agile manager game platform

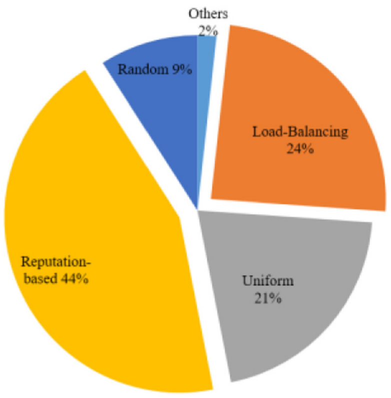

Figure 5. The distribution of human task allocation strategies

The reputation network is a three-layer BPNN and trained with the real task allocation strategies from the Agile Manager data set. The inputs to the network include the current reputation of the 10 worker agents and the strategies exhibited by the players for allocating the 30 tasks to the worker agents. Note that the tasks are 20 in total in some game levels; in that case, the remaining location in each task group is replaced by zero. The output of the network is the estimated reputation of ten worker agents in the immediate next time step. The inputs and outputs are normalised to the interval of [–1, 1] based on the minmax method described in equation (17); therefore, the tan-sigmoid function is used as the transfer function during the process of training the reputation network.

Note that 70 per cent of the real-world data are used to train the reputation network, and the remaining is used as the testing data. The performance of the reputation network with different number of neurons in hidden layer is shown as Figure 6. It can be observed that the mean square error (MSE) of the reputation network becomes lower as the number of neurons in the hidden layer increases. It means that the reputation network can learn more relationship between the inputs and outputs with increasing number of neurons in the hidden layer. The optimal number of neurons in hidden layer is 10. Increasing the number of neurons in the hidden layer further would result in negligible improvement in MSE but will incur additional costs in terms of training time.

With respect to the iteration between the policy and reputation network for obtaining a multi-round allocation strategy, we mainly study the performance of different iterations of our solution in the Agile Manager game platform. We select Game Level 6, which contains the most challenging task allocation scenario, to evaluate the proposed approach.

Figure 7 shows the comparison of the average score achieved in the game. In the data set, the average score achieved by the game AI is 1.5 times that achieved by human players. The histograms represent different settings of the total number of iterations between the policy network and reputation network in the proposed approach. As can be seen from Figure 7, when the number of iterations is 3 or more, the average score achieved by the proposed approach is always higher than the real human players' average score. This shows that the solution produced by the proposed approach through learning from the allocation strategies from human players is superior than average human performance. The performance of PNRN converges after five iterations of training. The ratio between the averages achieved by PNRN and by the players stabilizes around 54:46 (i.e. PNRN outperforms players by 17.4 per cent).

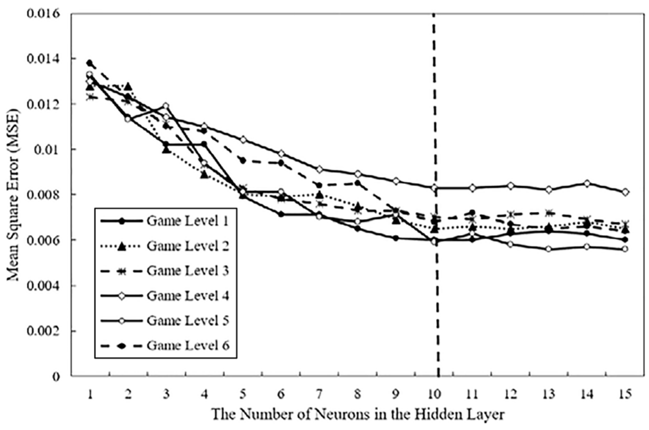

Figure 6.
The performance of the reputation network with different number of neurons in the hidden layer

We further compare the performance of PNRN trained with 5 iterations against the game AI in a round by round manner in Game Level 6 (Level 6 consists of 20 rounds in which 30 tasks must be allocated to 10 worker agents with different capabilities). Figure 8 shows the comparison results. Overall, PNRN outperforms the game AI which is detailed in (Yu *et al.*, 2013; Yu *et al.*, 2013). The ratio between the average score achieved by PNRN and that achieved by the game AI is 53:47 (i.e. PNRN outperforms the game AI by 12.7 per cent). The fluctuations in both approaches are caused by the information uncertainty associated with the worker agents' capabilities as both approaches need to estimate this information based on interaction experience within the 20 rounds of games.

6. Conclusions and future work

In this paper, we propose a learning approach combining supervised learning and reinforcement learning to enable agents to provide task allocation decision support in crowdsourcing by imitating the best observed human strategies. With the help of a unique large-scale real-world data set containing human task allocation decisions, we demonstrate that the proposed PNRN approach is effective in learning the from human task allocation strategies in multi-stage allocation of heterogeneous tasks to multiple workers. It is able to outperform both the human players and the advanced algorithmic crowdsourcing approach adopted by the game AI. PNRN can serve as a baseline approach to help the research community explore how to incorporate the knowledge embedded in the collective task allocation decisions made by crowdsourcing participants to arrive at efficient and effective computational task allocation strategies. It is a novel approach fulfilling an important gap in current research which mostly focuses on task allocation decision support mechanism

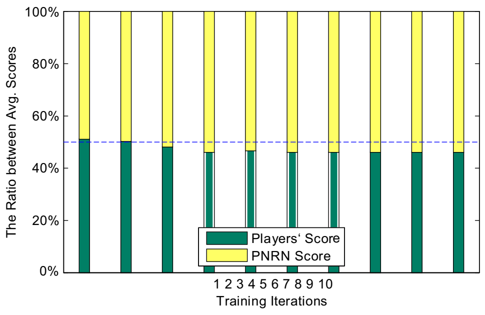

Figure 7. Comparison with human performance in Game Level 6

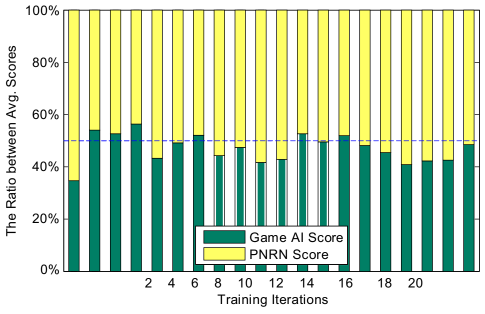

Figure 8. Comparison with game AI in Game Level 6

design based on expert knowledge and paves a possible way for simulating complex human behavioural dynamics in crowdsourcing environments (Galla and Farmer, 2013).

By only comparing with human behaviours in the data set based on which the PNRN model is trained and only against one AI-based approach, there is risk that the observed performance could potentially be affected by over-fitting. Thus, we refrain from drawing any conclusion about the general applicability of the PNRN approach. In our future research, we will investigate this question by studying the performance of PNRN in practical crowdsourcing systems.

In more complex application scenarios, workers may be able to obtain partial rewards for partially completed tasks. In future research, we will also incorporate reward models which can accommodate partial task completion situations into the learning framework. In addition, we will also consider to automatically and dynamically changing the number of tasks for allocation based on the actual demand.

References

Aljanaki, A., Wiering, F. and Veltkamp, R.C. (2016), "Studying emotion induced by music through a crowdsourcing game", *Information Processing & Management*, Vol. 52 No. 1, pp. 115-128.

Chen, K., Yang, S. and Batur, C. (2012), "Effect of multi-hidden-layer structure on performance of BP neural network: Probe", *Proceedings of the 8th International Conference on Natural Computation (ICNC'12)*, pp. 1-5.

Ho, C.J. and Vaughan, J.W. (2012), "Online task assignment in crowdsourcing markets", *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI-12)*, pp. 45-51.

Watkins, C.J.C.H. and Dayan, P. (1992), "Q-learning", *Machine Learning*, Vol. 8 Nos 3/4, pp. 279-292.

Galla, T. and Farmer, J.D. (2013), "Complex dynamics in learning complicated games", *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15) (AAMAS'15)*, Vol. 110 No. 4, pp. 1232-1236.

Goh, A.T.C. (1995), "Back-propagation neural networks for modeling complex systems", *Artificial Intelligence in Engineering*, Vol. 9 No. 3, pp. 143-151.

Jagabathula, S., Subramanian, L. and Venkataraman, A. (2014), "Reputation-based worker filtering in crowdsourcing", *Advances in Neural Information Processing Systems*, pp. 2492-2500.

Jøsang, A., Ismail, R. and Boyd, C. (2007), "A survey of trust and reputation systems for online service provision", *Decision Support Systems*, Vol. 43 No. 2, pp. 618-644.

Kartal, B., Nunes, E., Godoy, J. and Gini, M. (2016), "Monte Carlo tree search for multi-robot task allocation", *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*.

Khare, R., Good, B.M., Leaman, R., Su, A.I. and Lu, Z. (2015), "Crowdsourcing in biomedicine: challenges and opportunities", *Briefings in Bioinformatics*, Vol. 17 No. 1, pp. 23-32.

LaToza, T.D. and van der Hoek, A. (2016), "Crowdsourcing in software engineering: models, motivations, and challenges", *IEEE Software*, Vol. 33 No. 1, pp. 74-80.

Liu, S., Miao, C., Liu, Y., Yu, H., Zhang, J. and Leung, C. (2015*)*, "An incentive mechanism to elicit truthful opinions for crowdsourced multiple choice consensus tasks", *Proceedings of the 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pp. 96-103.

Liu, X., Lu, M., Ooi, B.C., Shen, Y., Wu, S. and Zhang, M. (2012), "Cdas: a crowdsourcing data analytics system", *Proceedings of the VLDB Endowment*, Vol. 5, 1040-1051.

Macarthur, K.S., Stranders, R., Ramchurn, S.D. and Jennings, N.R. (2011), "A distributed anytime algorithm for dynamic task allocation in multi-agent systems", *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-11)*.

Man, Z., Lee, K., Wang, D., Cao, Z. and Miao, C. (2011), "A new robust training algorithm for a class of single-hidden layer feedforward neural networks", *Neurocomputing*, Vol. 74 No. 16, pp. 2491-2501.

Miao, C., Yu, H., Shen, Z. and Leung, C. (2016), "Balancing quality and budget considerations in mobile crowdsourcing", *Decision Support Systems*, Vol. 90, pp. 56-64.

Nath, S. and Narayanaswamy, B.M. (2014), "Productive output in hierarchical crowdsourcing", *Proceedings of the 13th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS'14)*, pp. 469-476.

Tran-Thanh, L., Huynh, T.D., Rosenfeld, A., Ramchurn, S.D. and Jennings, N.R. (2014), "Budgetfix: budget limited crowdsourcing for interdependent task allocation with quality guarantees", *Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 477-484.

Tran-Thanh, L., Venanzi, M., Rogers, A. and Jennings, N.R. (2015), "Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks", *Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS'13)*, pp. 901-908.

Yu, H., Miao, C., An, B., Leung, C. and Lesser, V.R. (2013), "A reputation management approach for resource constrained trustee agents", *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'13)*, pp. 418-424.

Yu, H., Miao, C., An, B., Shen, Z. and Leung, C. (2014), "Reputation-aware task allocation for human trustees", *Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 357-364.

Yu, H., Miao, C., Leung, C., Chen, Y., Fauvel, S., Lesser, R.V. and Yang, Q. (2016), "Mitigating herding in hierarchical crowdsourcing networks", *Scientific Reports*.

Yu, H., Miao, C., Shen, Z. and Leung, C. (2015a), "Quality and budget aware task allocation for spatial crowdsourcing", *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1689-1690.

Yu, H., Miao, C., Shen, Z., Leung, C., Chen, Y. and Yang, Q. (2015b), "Efficient task sub-delegation for crowdsourcing", *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI-15)*, pp. 1305-1311.

Yu, H., Lin, H., Lim, S.F., Lin, J., Shen, Z. and Miao, C. (2015c), "Empirical analysis of reputation-aware task delegation by humans from a multi-agent game", *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15)*, pp. 1687-1688.

Yu, H., Shen, Z., Miao, C. and An, B. (2012), "Challenges and opportunities for trust management in crowdsourcing", *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'12)*, pp. 486-493.

Yu, H., Shen, Z., Miao, C. and An, B. (2013), "A reputation-aware decision-making approach for improving the efficiency of crowdsourcing systems", *Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS'13)*, pp. 1315-1316.

Yu, H., Shen, Z., Miao, C., Leung, C., Chen, Y., Fauvel, S., Lin, J., Cui, L., Pan, Z. and Yang, Q. (2017), "A dataset of human decision-making in teamwork management", *Scientific Data*, Vol. 4, p. 160127.

Yu, H., Yu, X., Lim, S.F., Lin, J., Shen, Z. and Miao, C. (2014), "A multi-agent game for studying human decision-making", *Proceedings of the 13th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS'14)*, pp. 1661-1662.

Further reading

Dan, C.C., Cires, C. and Meier, U. (2012), Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012), "Imagenet classification with deep convolutional neural networks", *Advances in Neural Information Processing Systems*, pp. 1097-1105.

Luz, N., Silva, N. and Novais, P. (2015), "A survey of task-oriented crowdsourcing", *Artificial Intelligence Review*, Vol. 44 No. 2, pp. 187-213.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M. (2012), Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V. and Lanctot, M. (2012), "Mastering the game of go with deep neural networks and tree search", *Nature*, Vol. 529, pp. 484-489.

Trivella, A. and Pisinger, D. (2016), "The load-balanced multi-dimensional bin-packing problem", *Computers & Operations Research*, Vol. 74, pp. 152-164.

Whang, S.E., Lofgren, P. and Garcia-Molina, H. (2013), "Question selection for crowd entity resolution", *Proceedings of the VLDB Endowment*, Vol. 6 No. 6, pp. 349-360.

Yu, H., Shen, Z., Miao, C., Leung, C. and Niyato, D. (2010), "A survey of trust and reputation management systems in wireless communications", *Proceedings of the IEEE*, Vol. 98 No. 10, pp. 1755-1772.

Corresponding author
Xudong Zhao can be contacted at: sdu_zxd@163.com